

HACKberry

HANDBOOK

User manual to assemble and customize HACKberry



目次

1. HACKberryとは

- 01 HACKberryの目指す世界 P006
- 02 仕様..... P008
- 03 ライセンス形態 P010
- 04 保障・責任に関して P013

2. 組み立ててみよう

- 01 どのHACKberryを組み立てますか P016
- 02 部品の入手方法 P019
- 03 組み立てる前に P022
- 04 基本のはんだ付け P026
- 05 プログラムのアップロード P052
- 06 バッテリーケース・ソケットの組み立て P058
- 07 親指の組み立て P074
- 08 人差し指の組み立て P080
- 09 三指の組み立て P088
- 10 電圧の調整とサーボモーターの初期位置設定 P092
- 11 掌の組み立て P094
- 12 手首の組み立て P122
- 13 センサの組み立て P130
- 14 初期位置設定 P134

3.動かしてみよう

- 01 各部名称 P146
- 02 バッテリーの取り付け P147
- 03 キャリブレーション P148
- 04 手の取り付け P150
- 05 手首の撓尺屈 P151
- 06 筋電センサでの動作 P151

4. HACKberryをもっと楽しもう

- 01 はじめに P154
- 02 HACKberry hand board Mk2 仕様 P155
- 03 筋電センサによるHACKberryの操作 P158
- 04 アセトン表面処理 P162
- 05 ナイロン染色 P168

5.故障かな?と思ったら P174

1. HACKberryとは

01 HACKberryの目指す世界	P006
02 仕様	P008
03 ライセンス形態	P010
04 保障・責任に関して	P013

Introduction

01 HACKberryの目指す世界



【1】HACKberryとは何か？

HACKberryはexiii株式会社が設計し2015年に発表した電動義手です。その筐体は3Dプリントにより製造され、また、全設計データがオープンソースとしてweb上に公開されています。2016年9月以降は、特定非営利活動法人Mission ARM Japanに引き継がれ、現在も改良および普及活動が進められています。

電動義手とは、筋肉の電気信号などを介して操作する電動の義手のことです。筋電センサなどを皮膚表面にあて、計測される信号のパターンから使用者の意図を読み取り、これに応じて義手に内蔵されたモーターが動きます。使用者は腕に少し

力を入れるだけで、健常者と同じように直感的に義手を動かせます。

電動義手の開発の歴史は長く、1960年代にはすでに市販化されています。しかし、従来の製造および流通の仕組みの中では価格を下げるのが難しく、未だに150万円以上ととても高価です。そのため、電動義手を入手できるのは先進国の一部の人たちに限られます。

HACKberryは3Dプリントとオープンソースを組みわせることで、電動義手を「気軽な選択肢」とすることを目指しています。3Dプリントは金型代などの初期費用がかからず、また、webを介してオープンソースのデータにアクセスすればその場ですぐに作り始めることができます。HACKberry



HACKberryの世界への広がり

はこのように製造や流通に関わるコストを抑え、5万円程度の材料費を払えば、世界中どこにいても電動義手を自作できる仕組みをつくりました。

まだ市販品のような実用的な品質には達していませんが、世界中で様々な方たちがHACKberryを作り始めています。



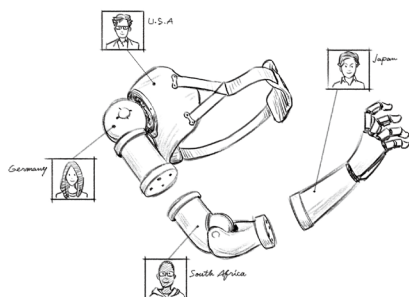
義手のカスタマイズ

【2】HACKberryのビジョン

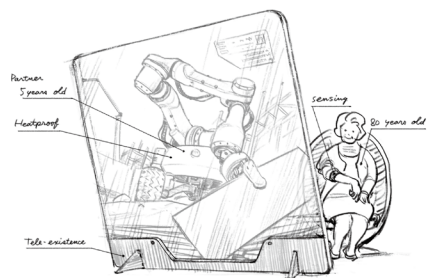
HACKberryのビジョンは「気軽な選択肢」です。これは「価格」および「デザイン」という二つの側面に関して掲げています。上述した通り、現状の電動義手はとても高く、価格が下がらなければ広く普及しないでしょう。ただし、価格を下げるだけではなく、デザインの幅を広げることも大切です。わたしたちの趣味や生活スタイルは千差万別です。これまで義手は福祉機器として生身の手に似せたデザインが一般的でしたが、義手を靴、洋服、メガネなどと同じようにファッションの一部と考えると、義手のデザインの可能性も広がります。HACKberryはその一例として、敢えて肌色にしないスタイリッシュなデザインが施されています。

また、義手を「気軽な選択肢」にしていくためには、「使う人」だけでなく「作る人」にも焦点を当てなければいけません。義手の使い手と作り手は表裏一体の関係にあります。義手を使うチャンスを増やすためには、義手を作るチャンスを増やしていかなければいけません。そして、義手を作るチャンスを増やすためには「作る楽しさ」を伝えていくことが大切だと考えています。例えば料理をつくるように、義手を使う人のライフスタイルを想像しながら、創意工夫をこらして様々なデザインの義手が次々とつくられるような世界を目指しています。

このようなビジョンのもとにHACKberryは名付けられました。Hackberryはニレ科の広葉樹であるエノキの英名です。エノキは広く枝を伸ばしてたくさんの葉と実をつけます。HACKberryも、「改良(HACK)することで楽しみ方の枝葉が広がり、ユーザーの手に届き実(berry)を結ぶ」という思いを込め、世界中で様々な義手を実現させる大樹に育てていきたいです。



オープンソースコミュニティによる開発



義手以外の応用



葉をつまむ HACKberry

【1】HACKberryの仕組み

まず、HACKberryの標準的な仕様を次ページの表に示します。寸法および重量は手部に関するものであり、ソケット部分は含まれません。

HACKberryで利用するセンサは簡易圧力センサー一つです。このセンサは、フォトフレクタとよばれる安価な距離センサに、内部に空洞のある10mm厚のスポンジを重ねて、作成します。センサをユーザーの腕の上(なるべく筋肉のある部分)に置き、これを非伸縮性バンドで腕のまわりに固定することで、ユーザーの腕の筋肉の動きを物理的に読み取ることができます。(この技術は大阪工業大学の吉川雅博先生が開発した技術を参考にしています。)

3つあるサーボモーターの角度は簡易圧力センサの読み取る値によって変化します。このときの情報処理はマイコンが行います。ユーザーが腕に力を入れると手先は閉じ、力を抜くと手先は開きます。また、入れる力が強いほど手先の閉じるスピードは速くなります。オープンソースとして公開しているプログラムはこのような設定になっていますが、プログラムを書き換えて違う設定にすること(例えば、手先の開閉を逆転させるなど)も可能です。

【2】HACKberryの設計上の特徴

いまの科学技術では生来の巧みな手を完璧に再現することは難しく、義手を設計する際には機能の取舍選択が求められます。HACKberryでは日常生活における実用性を念頭に置き、下記のような特徴を持たせました。

(1)「つまむ」と「握る」の両立

人は手にものを取るとき、対象となるものの形に合わせて指や手首の姿勢を巧みに調整することができます。しかし義手の場合、モーターの数に限られるため、生来の手のように自由な調整は行えません。そこで、HACKberryは人指し指に次

ページ図のようなリンク機構を採用することで、小さなものを「つまむ」機能と大きなものを「握る」機能を両立させています。このリンク機構では、ものが指の中腹に当たると付け根の関節の回転力が先端の関節に送られるため、モーター1つで、小さいものに対してはつまみ、大きいものに対しては握ることを受動的に切り替えられます。

(2) 組み立てやすさ

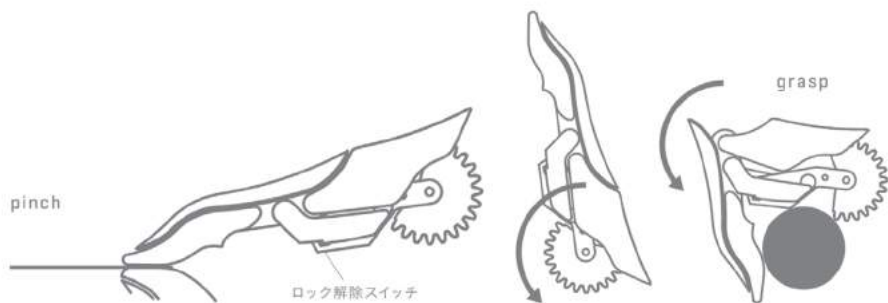
義手は一般的な電化製品と比べて複雑な機構をしています。また、多様な環境の中で使われるため故障するリスクも高いです。さらに、HACKberryはオープンソースとして第三者が組み立てられることを前提としています。これらの理由から、HACKberryが組み立てやすく、また、組み直しやすいことは重要です。HACKberryの構成部品はすべて世界中で入手可能なものを使っています。また、指関節の駆動にワイヤを用いずギヤを採用しました。ワイヤは腱に似た柔らかい動きを表現しやすいですが、逆に緩みやすくもあり、頻繁にメンテナンスが必要です。一方、ギヤだと固い動きになってしまいますが、緩むことはなく故障した際にも修理しやすいです。

(3) 手首機構

日常生活における利便性を考えると、手先だけでなく手首の動きも軽視できません。HACKberryの手首は手動で角度を調整できます。手首部分に配置されたボタンを押すと、手首を回転させたり、傾けたりすることができます。また、手部と手首部を着脱し、配置を90度ごとに調整できます。これらを組み合わせると手首が持つ3つの自由度(回内/回外、掌屈/背屈、橈屈/尺屈)を再現できます。詳しくは、「3.HACKberryをもっと楽しもう」(P145)をごらんください。

寸 法	225 x 150 x 60 (mm) ※親指が並立位にある場合
重 量	450～500g ※材料により変動
セ ン サ	簡易圧力センサ x1
モ ー ター	サーボモーター(大) x1 ※人差し指の屈曲 サーボモーター(小) x2 ※親指の内外転、中/薬/小指の屈曲
バッテリ	Li-ion電池 (7.2V, 2200mAh)
マイコン	Mk1: Arduino micro Mk2: Arduino nano互換品

HACKberry仕様



HACKberryの人差し指のリンク機構

Introduction

03 ライセンス形態

オープンソースとしてHACKberryを利用される場合、exiii株式会社が定めた以下の規約に従う必要があります。

====

HACKberry Open Source Project Policy

イクシー株式会社(以下「当社」といいます。)は、本ポリシーにおいて、「HACKberry」に関するソースコードおよびデータ等の公開(以下「本プロジェクト」といいます。)の内容およびその利用条件等について定めます。

1.目的

1.1. 本プロジェクトの目的は、当社が本プロジェクトの一切の利用者(以下「ユーザー」といいます。)に対し、本ポリシーに基づいてソースコードおよびデータを公開し、当社の発明について利用許諾をすることで、義手の研究開発および普及を促進することにあります。

1.2. 本ポリシーに基づいて公開されたソースコードおよびデータ、または本ポリシーに基づいて利用許諾された発明について、以下の各号に規定する目的での使用を禁止します。

- 1.2.1. 軍事目的での使用
- 1.2.2. 公序良俗に反する目的での使用
- 1.2.3. 当社またはHACKberryの名誉または信用毀損する目的での使用
- 1.2.4. 当社が上記各号に該当するおそれがあると判断する目的による使用

2. ソフトウェアのソースコード

HACKberryに関するソフトウェアのソースコードは、以下のライセンスに基づき公開されます。

各ライセンスとは異なる条件(商用利用等)でソースコードの利用を希望する場合は、以下のアドレスまでご連絡ください。

[info@exiii.jp](mailto:info@exiii.jp)

ソースコード ライセンス

2.1. Arduinoスケッチ(義手に内蔵したArduino Microに書き込むためのソースコードを意味します。) GNU General Public License version 3

3. ハードウェアに関するデータ

HACKberryに関するハードウェアに関する各データは、以下の各ライセンスまたは条件に基づき公開されます。

各ライセンスとは異なる条件(商用利用等)でデータの利用を希望する場合は、以下のアドレスまでご連絡ください。

[info@exiii.jp](mailto:info@exiii.jp)

データ ライセンス

3.1. 3Dデータ(義手に使う各部品の形状データを意味します。) Creative Commons License BY-NC-SA 4.0 International

3.2. ハンド基板データ(マイコン、モーター、センサおよび電池を繋ぐための回路基板を意味します。) Creative Commons License BY-NC-SA 4.0 International

3.3. センサ基板データ(筋肉の隆起を検出するための反射型光センサ(フォトリフレクタ)を取り付けるための回路基板を意味します。) Creative Commons License BY-NC-SA 4.0 International

3.4. 電池基板データ(電池ボックスに内蔵する回路基板を意味します。) Creative Commons License BY-NC-SA 4.0 International

4. 特許

4.1. 当社は、ユーザーに対し、2.および3.1に定めたライセンスに基づいてソースコードまたはデータを利用して義手を制作するに当たって必要な範囲であってかつ1.2.に違反しない場合に限り、当社が特許出願中または特許登録済みの発明を実施することを、無償で、地域または期間の限定なく、非独占的に許諾（再許諾は除きます。）します。

4.2. 4.1.に定める発明は、関節機構に関するものであって、簡単な構造で人間の手の動作に近い動作を実現可能とするものです。現在特許出願中ですが、出願手続の進展に応じて、発明の特定に関する情報を本ポリシーまたはフォーラム等において適切な方法により告知します。

5. 商標等およびコンテンツ

5.1. 以下の商標は、当社が日本において登録済の商標です。

商標: exiii

登録番号: 5786270

5.2. 以下の商標は、当社が日本において出願中の商標です。

商標: HACKberry

出願番号: 2016-042844, 2016-042845

5.3. 本ポリシーに特に定めがない限り、exiiiに関する商標等およびコンテンツについて、著作権（著作権法第27条および第28条の権利その他の権利を含みます。以下、同様とします。）、特許権、実用新案権、意匠権、商標権等の知的財産権（それらの権利を取得し、またはそれらの権利につき登録等を出願する権利も含むものとします。以下、総称して「知的財産権」といいます。）その他一切の権利は、当社に帰属します。

5.4. ユーザーが本ポリシーに基づいて公開されたソースコードおよびデータに関し、GitHubまたはフォーラム (<http://exiii-hackberry.com/forums/>) 上でコミットまたはアップロードした場合、その内容が著作権法上の著作物であるときには、コミットと同時に、ユーザーの著作権は当社に移転します。また、この場合、ユーザーは、当社または当社が許諾する第三者に対し、著作人格権を行使しないものとします。

5.5. 商標等およびコンテンツは、当社の事前の書面による許諾を得ない限り、いかなる利用もしてはなりません。

5.6. 商標等またはコンテンツの利用を希望する場合は、以下のアドレスまでご連絡ください。

<連絡先>info@exiii.jp

6. 個別のライセンスの優先

第2項または第3項に規定するソースコードまたはデータに付与されたライセンスに規定する事項のなかで、本ポリシーと第2項または第3項に規定するソースコードまたはデータに付与されたライセンスとの間に、同一の事項について異なる内容がある場合には、特段の定めがないかぎり、各ライセンスの内容を優先して適用する。

7. 免責

7.1. 当社は、本ウェブサイト、本ウェブサイトで提供されるソースコードおよびデータならびに本ポリシーに掲げる各ライセンスに事実上または法律上の瑕疵（安全性、信頼性、正確性、完全性、有効性、特定の目的への適合性、セキュリティなどに関する欠陥、エラーやバグ、権利侵害などを含みます。）がないことを明示的にも黙示的にも保証しておりません。当社は、ユーザーに対して、かかる瑕疵を除去してHACKberryに関するサービスを提供する義務を負いません。

7.2. 当社は、当社が提供するソースコードおよびデータの使用または本ポリシーに定める各ライセンスの解釈適用に関連してユーザー間またはユーザーと第三者との間で生じた、いかなる形態での紛争またはあらゆる損害について一切の責任を負いません。

7.3. 当社は、本ウェブサイトにおいてソースコードおよびデータが常に提供されることを保証しません。当社は、本ウェブサイトのシステムの保守点検といった理由をとわず、本ウェブサイトにおいてソースコードおよびデータが提供できない事態が発生したことによってユーザーまたは第三者に生じるあらゆる損害について一切の責任を負いません。

8. 提供方法および内容の変更

8.1. 当社は、ユーザーに対してあらかじめ通知することなく、いつでも本ポリシーに定めるソースコードもしくはデータの提供方法もしくは提供内容または発明の利用許諾の内容を変更し、または提供もしくは利用許諾を中止すること(以下「変更等」といいます。)ができます。

8.2. 当社は、前項の変更等に関連して、ユーザーまたは第三者に生じるあらゆる紛争または損害について一切の責任を負いません。

9. 本ポリシーの変更

9.1. 当社は、ユーザーに対してあらかじめ通知することなく、いつでも本ポリシーの内容を変更することができます。

9.2. ユーザーは、本ポリシーの変更があった後、当社が提供するソースコードもしくはデータの取得もしくは利用を継続することによって、または当社の発明の利用を継続することによって、変更後の本ポリシーについて有効かつ取消不能な同意をしたものとみなします。ただし、本ポリシー変更前にユーザーが変更前のポリシーに基づいて取得したソースコードまたはデータに適用されるライセンスの内容はこの限りではありません。

10. 準拠法、裁判管轄

10.1. 本ポリシーは、日本語を正文とし、準拠法は日本法とします。

10.2. 本ポリシーに起因し、または関連する一切の紛争については、訴額に応じて、東京地方裁判所または東京簡易裁判所を第一審の専属的合意管轄裁判所とします。

====

以上がHACKberryをオープンソースとして利用する場合に従うべき規約になります。なお、HACKberryに関連して、exiii-hackberry.com上のフォーラムやGitHubにアップロードされた内容は、プログラムコードに限らず、不具合報告、意見、アイデア、その他内容の如何や、営利または非営利利用を問わず、「HACKberry」の開発に利用され、Creative Commonsライセンスのみならず、exiii株式会社が独自に設定するクローズドソースライセンスの下で頒布されることがあります。

Introduction

04 保障・責任に関して

HACKberryは3Dプリンタおよびオープンソースを活用して、世界中の誰もが簡単に電動義手を自作できるようにしました。HACKberryをこのような形で提供する理由は、義手を使う人が電動義手を試せ、義手を作る人がその楽しさを知れる機会を少しでも増やしたいと考えるからです。

しかしながら、HACKberryは市販の電動義手のような十分な安全性は確保されていなく、また、完成品としての販売・品質保証を行っておりません。研究や試用の範囲を越えてHACKberryを義手として使う際は、HACKberryの性能や精度を十分理解し、自己責任のもと行なって下さい。HACKberryを使用してユーザーまたは第三者に生じるあらゆる紛争または損害について、exiii株式会社および特定非営利活動法人Mission ARM Japanは一切の責任を負いません。

2. 組み立ててみよう

- 01 どのHACKberryを組み立てますかP016
- 02 部品の入手方法P019
- 03 組み立てる前にP022
- 04 基本のはんだ付けP026
- 05 プログラムのアップロードP052
- 06 バッテリーケース・ソケットの組み立てP058
- 07 親指の組み立てP074
- 08 人差し指の組み立てP080
- 09 三指の組み立てP088
- 10 電圧の調整とサーボモーターの初期位置設定P092
- 11 掌の組み立てP094
- 12 手首の組み立てP122
- 13 センサの組み立てP130
- 14 初期位置設定P134

01 どのHackberryを組み立てますか

HACKberryにはいくつかの構成が存在します。まずはどのHACKberryを組み立てるかを決めましょう。

A. 右手



最もベーシックなタイプのHACKberryです。義手として使用する際は実際の腕に適合するソケットを義肢製作所などに依頼して作ります。本イメージとは別に、HACKberryを操作するためのセンサと電源を共有するための外部バッテリーケースが必要になります。

B. 右手(ダミーソケット付き)



AのHACKberryに3Dプリントされた仮のソケットが付いたタイプです。ソケット内部にバッテリーを格納するので外部バッテリーケースが不要です。本イメージとは別にHACKberryを操作するためのセンサが必要になります。

C. 左手



最もベーシックなタイプのHACKberryの左手仕様です。義手として使用する際は実際の腕に適合するソケットを義肢製作所などに依頼して作ります。基本的には右手との差異はありません。本イメージとは別に、HACKberryを操作するためのセンサと電源を共有するための外部バッテリーケースが必要になります。

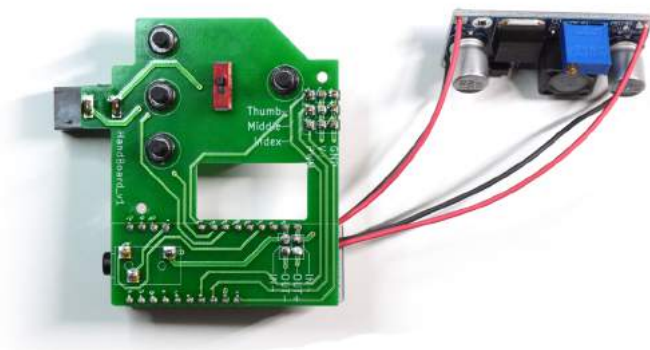
D. 左手 (ダミーソケット付き)



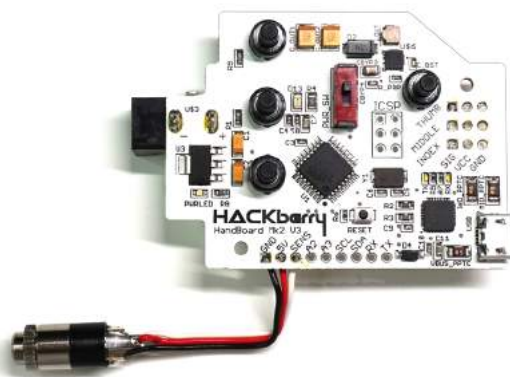
CのHACKberryに3Dプリントされた仮のソケットが付いたタイプです。ソケット内部にバッテリーを格納するので外部バッテリーケースが不要です。本イメージとは別にHACKberryを操作するためのセンサが必要になります。

また、HACKberryにはMk1とMk2の2種類の基板が存在します。どちらの基板でもHACKberryは動作しますが、Mk2の方が実装が簡易で、低価格です。同時に基板自体のサイズが小さく、機能拡張用のポートが用意されているのでカスタマイズ性に優れています。

•HACKberry hand board Mk1



•HACKberry hand board Mk2



02 部品の入手方法

【1】3Dプリント部品の印刷

3Dプリント部品のデータはインターネットから入手することができます。以下のリンクを開いて表示されたページのDATA(GITHUB)をクリックしてください。

<http://exiii-hackberry.com/>



Githubのサイトが開きます。

<https://github.com/mission-arm/HACKberry>

そこから

HACKberry_3Dmodel / STL / ...

と進み、目的のSTLデータをダウンロードします。

The image shows a screenshot of the GitHub repository page for 'mission-arm / HACKberry'. The repository has 108 watches, 307 stars, and 133 forks. The current branch is 'master'. The file path is 'HACKberry / HACKberry_3Dmodel / STL /'. The commit history is as follows:

Commit	Message	Time
sktomtomoto	add STL and STEP data for SLS print	Latest commit 922f7d4 4 days ago
C	upgrade hardware, circuits and software to mk2	5 days ago
ForSLSPrint	add STL and STEP data for SLS print	4 days ago
L	upgrade hardware, circuits and software to mk2	5 days ago
R	upgrade hardware, circuits and software to mk2	5 days ago

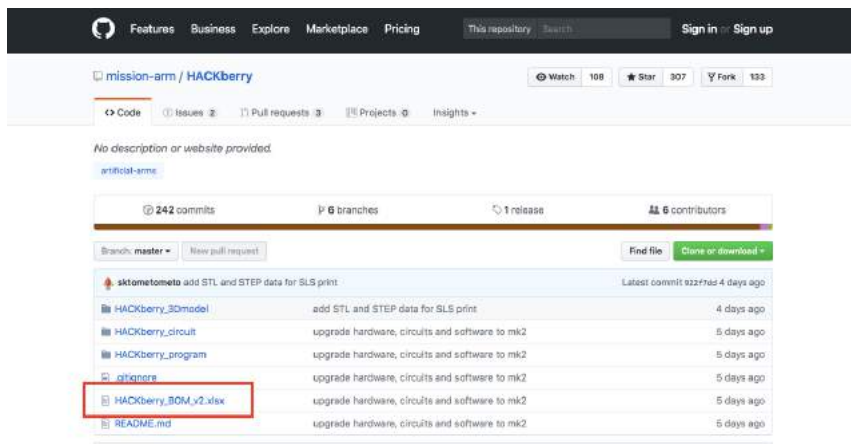
【2】3Dプリント部品以外の機械・電子部品の購入方法

まず前項を参考にしてGithubページにアクセスしてください。

<https://github.com/mission-arm/HACKberry>

そのページにある“HACKberry_BOM_v2.xlsx”をダウンロードします。

(BOMのバージョンは今後更新される可能性があります。)



ダウンロードしたファイルをExcelまたは互換ソフトを用いて開きます。

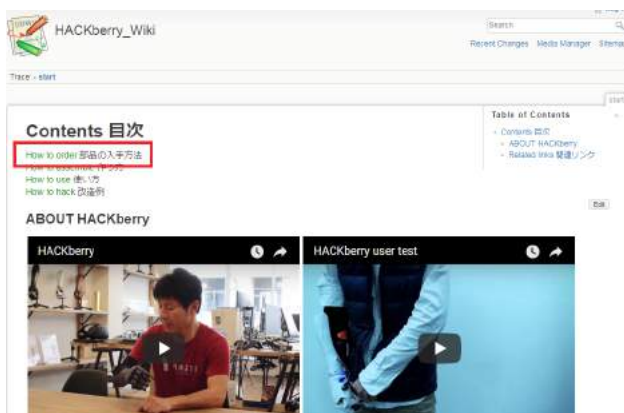
The screenshot shows an Excel spreadsheet with the following data:

Unit	Part Name	Description	Quantity	US delivery	US substitution	JP delivery	JP substitution
	SBI-SU-1e4	Steel ball -Spherical diameter:1/4"	2	Amazon.com		Amazon.jp	
	FlatScrewM2L6PHI3.3T1.3	Tapping screw M2L6 -Shaft diameter:2 -Head diameter:3.3 -Length:6	25			velco	
	ScrewM3.83t8e	Tapping screw M3L8 -Shaft diameter:3 -Head diameter:5.3 -Length:8	6			velco	
Hand	ES08MD	Servo motor(small)	2	Emga	Emga	KK HCRBY	Emga
	HD3001HB	Servo motor(large)	1	rcMarT		rcMarT	skruel
	RXF050	Resettable Fuses	3	Wuyser		Akiyuki	
	UsmBearing	Bearing -Bore diameter:4 -Outside diameter: -Width	1	spbusa		KMT	
	FlatScrewM2L10	Tapping screw M2L10 -Shaft diameter:2 -Head diameter:3.5 -Length:10	25			velco	

開いたファイルの画面右側のリンクから購入先のサイトを開くことができるので、部品名称、必要個数を参考にして、部品を購入してください。

HACKberryに使用する基板の入手方法に関しては、Mission ARM Japanにお問い合わせください。
(Mission ARM Japanを介さずに、回路図を参考にして自作するか、Seeed Studioなどのプリント基板製造サービスを用いて作成して頂くことも可能です。)詳しくは、以下のURLにアクセスし、TUTORIALをクリックして開かれたページを参照してください。

<http://exiii-hackberry.com/>



Assembling

03 組み立てる前に

注意事項

- ・組み立てはマニュアルの順番通りに行ってください。
- ・以降の説明は基本的に右手に関して行います。左手を組み立てる場合は画像を反転して考えてください。
- ・バージョンによっては本書のイメージと、お手元の部品に若干の違いが存在することがあります。

— 必要な工具 —

- | | | |
|----------------|---|----------|
| 1. はんだごて&こて台 | 2. 糸はんだ | 3. PC |
| 4. マイクロUSBケーブル | 5. ニッパー | 6. ドライバー |
| 7. ピンバイス | 8. ドリル($\phi 1.6\text{mm}$, $\phi 2.2\text{mm}$) | 9. メジャー |
| 10. テスター | 11. ビニールテープ | |



—あったら便利な工具・機材—

12. ラジオペンチ

13. ピンセット

14. ワイヤストリッパー

15. 熱収縮チューブ



⑫



⑬



⑭



⑮

—汎用部品—

1. ねじ(M2L10)

2. ねじ(M2L6)

3. ねじ(M3L8)

4. ベ어링

5. ワッシャー

6. トーションばね

7. コイルばね

8. 引張ばね

9. スペーサー

10. シャフト(10mm)

11. シャフト(15mm)

12. 鋼球

①

②

③

④

⑤

⑥



⑦

⑧

⑨

⑩

⑪

⑫



— 部品番号の見方 —

3Dプリント部品の命名規則について説明します。ハイフン(-)によって文字が区切られています。

第1項目は大分類です。

R	右手部品
L	左手部品
C	左右共用部品

第2項目はユニットの分類です。

SO	ソケットユニット
T	親指ユニット
I	人差し指ユニット
O	三指(中指、薬指、小指)ユニット
H	掌ユニット
W	手首ユニット
B	バッテリーケースユニット
SE	センサユニット

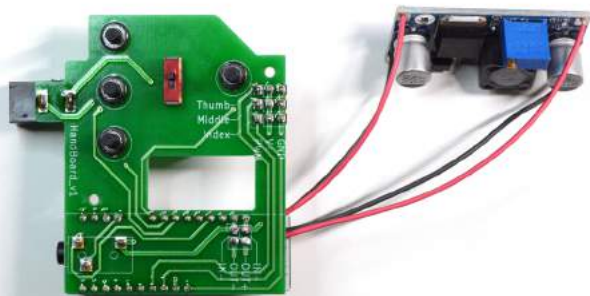
第3項目はそのユニット内で割り振られる番号です。

例として、R-SO-01 ならば、右手用ソケットユニットの1番部品ということになります。

Assembling

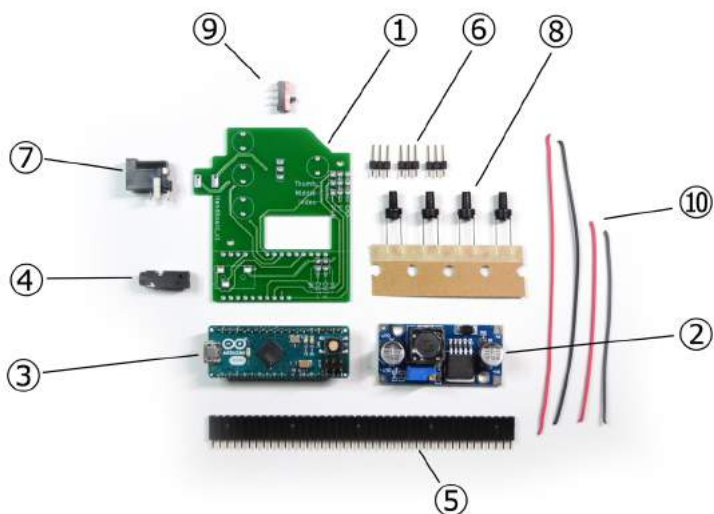
04 基板のはんだ付け

HACKberry hand board Mk1 のはんだ付け



—使用部品—

- | | | |
|-----------------------------|---------------|------------------|
| 1. HACKberry hand board Mk1 | 2. DCDCコンバーター | 3. Arduino Micro |
| 4. ステレオジャック | 5. ピンソケット | 6. ピンヘッド |
| 7. DCジャック | 8. タクトスイッチ | 9. スライドスイッチ |
| 10. リード線 | | |



*はんだ付けの際は、電気素子(部品)をしっかりと奥まで挿入し、浮いたり、斜めになったままはんだ付けをしないように注意してください。

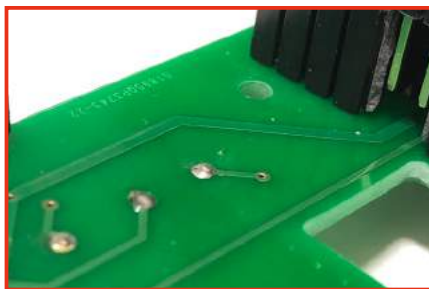
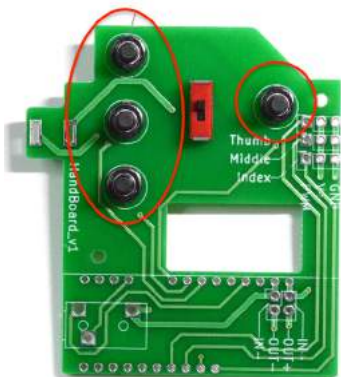
*基板に乗っている小さな電気素子は静電気で破損する可能性があります。はんだ付けをする前に静電気除去パッド等を使って静電気を可能な限り除去してください。

*水に濡れた手で電気素子に触れると回路がショートして破損の原因になります。はんだ付け作業はしっかりと水気を拭き取ってから行ってください。

【1】スライドスイッチを基板のオモテ側にはんだ付けします。基板は白い文字が印刷されている面が表面です。またスライドスイッチに極性(向き)はありません。

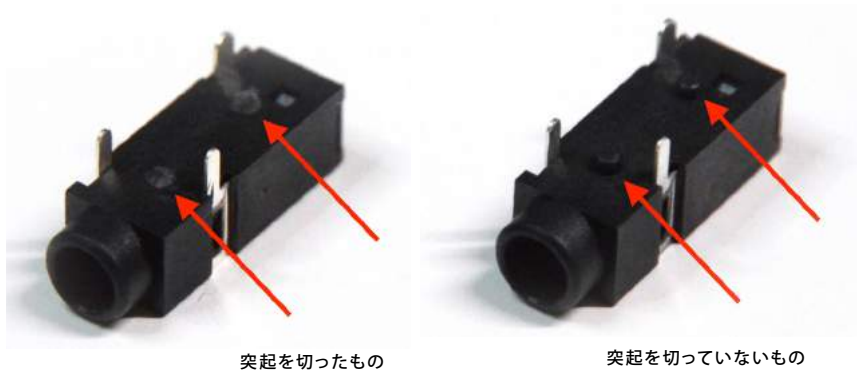


【2】タクトスイッチを基板のオモテ側にはんだ付けします。裏側に突き出て余ったリード線はニッパーでカットします。目安として基板から2mm以上突き出ないようにします。

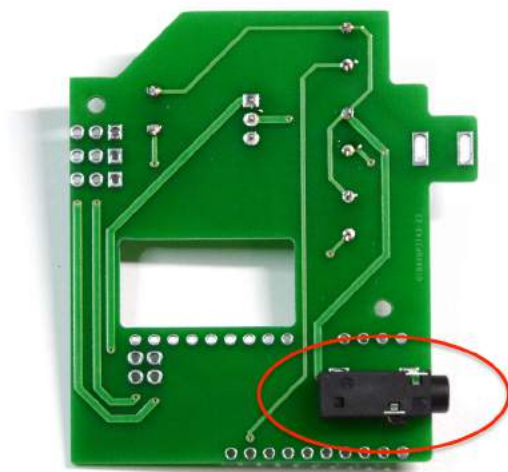


裏側

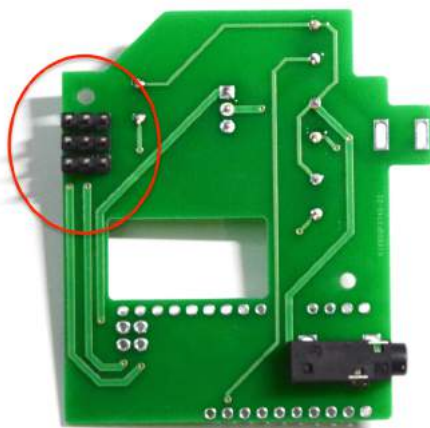
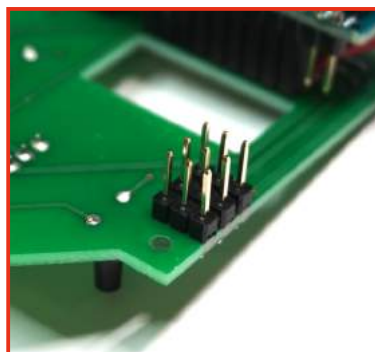
【3】次に裏側のはんだ付けをします。右手用の基板をはんだ付けしている場合、回路を作る前にステレオジャックの底面の突起をカッター等で切り取ってください。



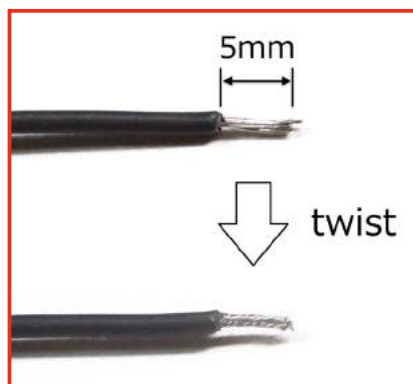
【4】次にステレオジャックを写真の位置にはんだ付けします。



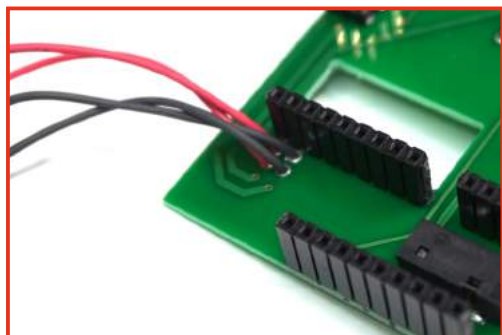
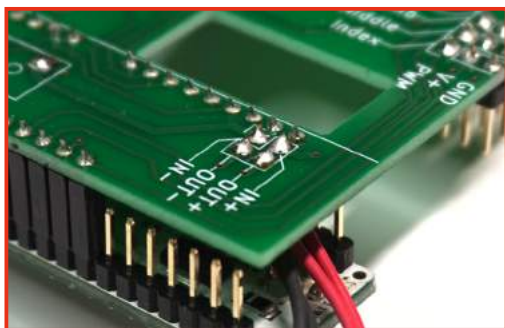
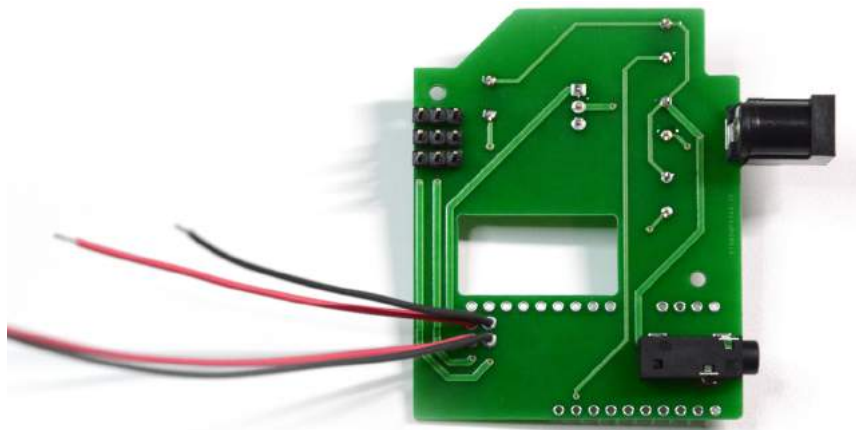
【5】ピンヘッダは**金属部分の短い方**を基板に挿してはんだ付けします。**ピンが斜めにならないように**気をつけます。



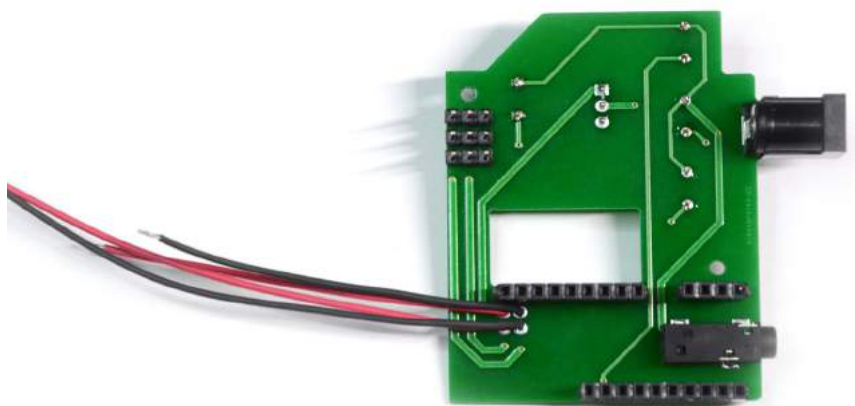
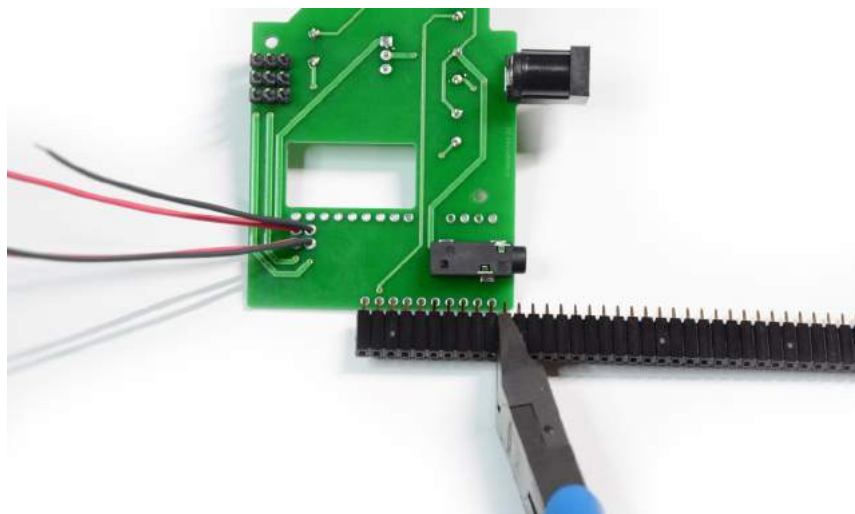
【6】リード線の先5mmの被覆をワイヤーストリッパー等を用いて剥き、線をねじってまとめます。ワイヤーストリッパーがない場合はニッパーで被覆を挟み、芯線を切らないように注意しながら被覆を剥きます。



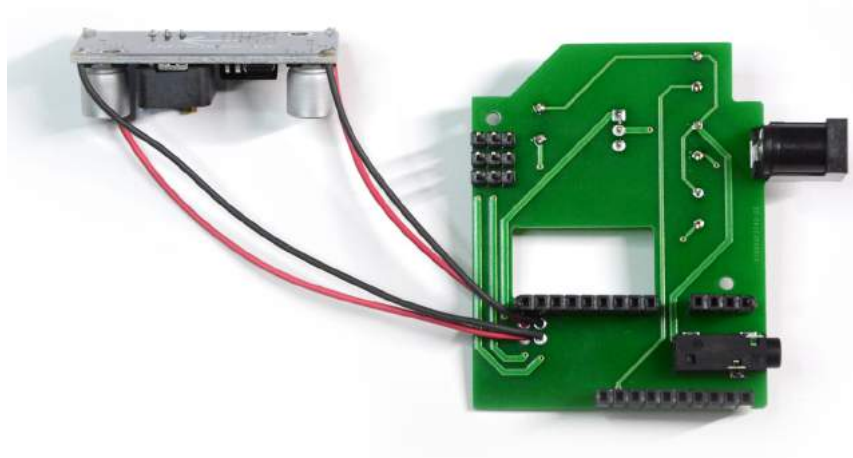
【7】長い方がOUTPUT,短い方がINPUTで、それぞれ赤が+に、黒が-につながるように基板にはんだ付けします。DCジャックもはんだ付けします。



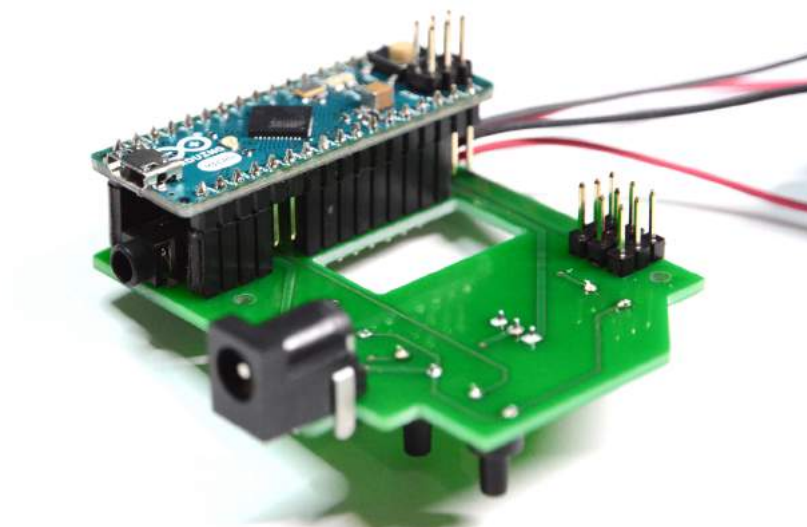
【8】基板の穴に合わせてニッパーでピンソケットをカットしてから3ヶ所はんだ付けします。



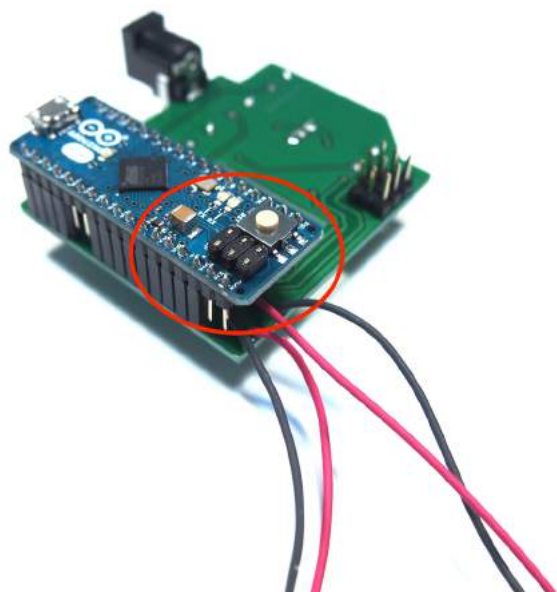
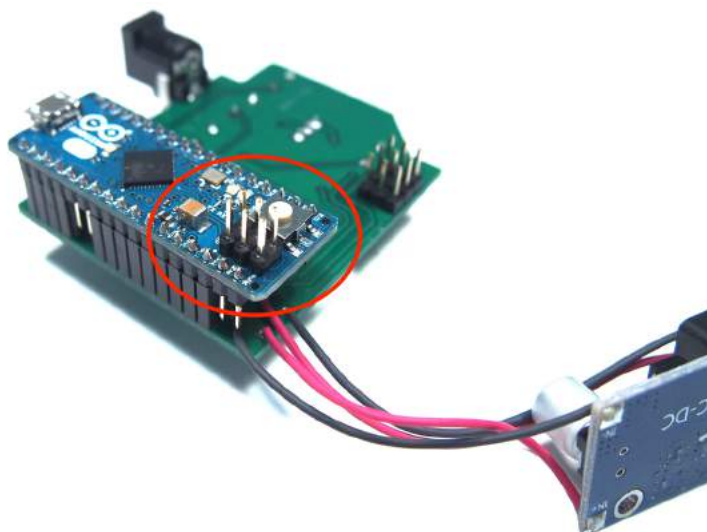
【9】DCDCコンバーターとHACKberry hand board Mk1のOUT+,OUT-,IN+,IN-を対応させてはんだ付けします。



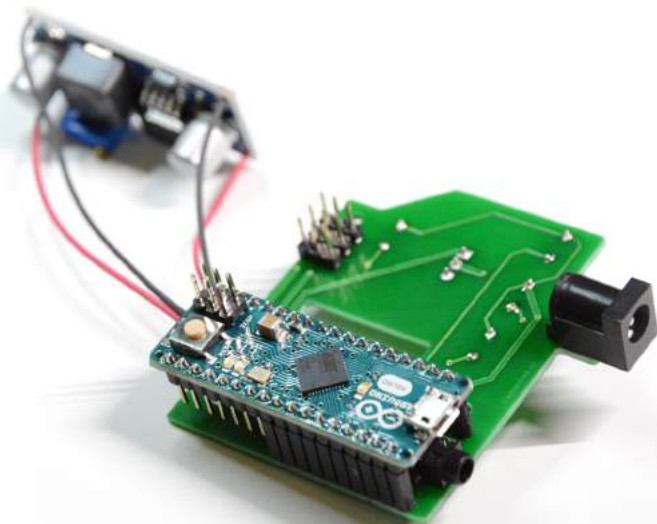
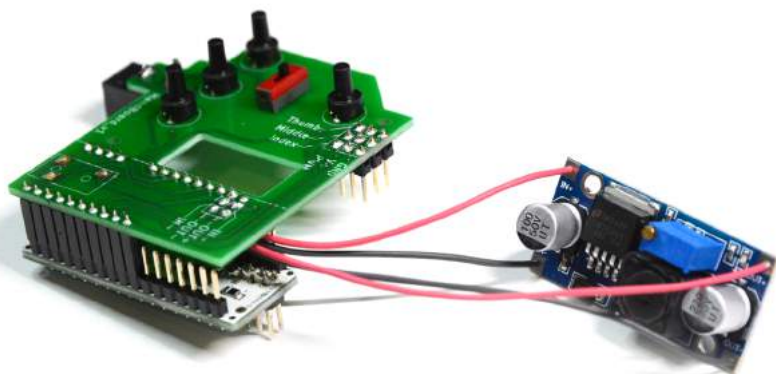
【10】Arduino Microを穴の位置を間違えないようにピンヘッダに奥まで挿し込みます。



【11】左手基板の場合は、図の赤枠内にあるピンを写真のようにニッパーでカットしてください。



【12】これでHACKberry hand board Mk1は完成です。



【13】次にサーボモーターの内部にポリスイッチを挿入していきます。

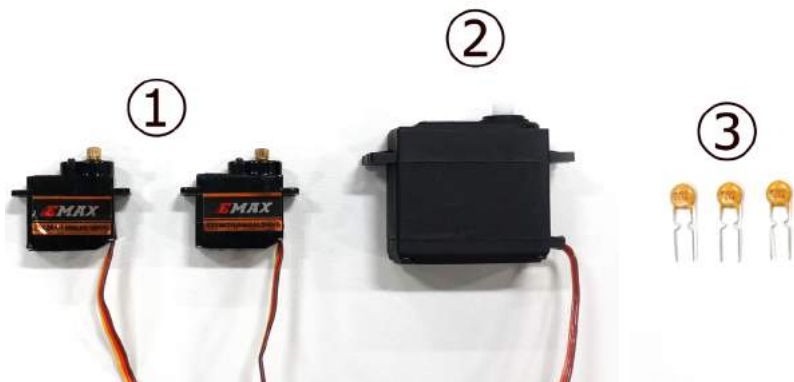
※この工程を省略してもサーボモーターを動かすことは可能ですが、ポリスイッチを取り付けることでモーターに過度の負荷がかかった時の破損を防ぐことができます。

※Mk2基板には基板自体に既にポリスイッチが実装されているのでこの作業は不要です。

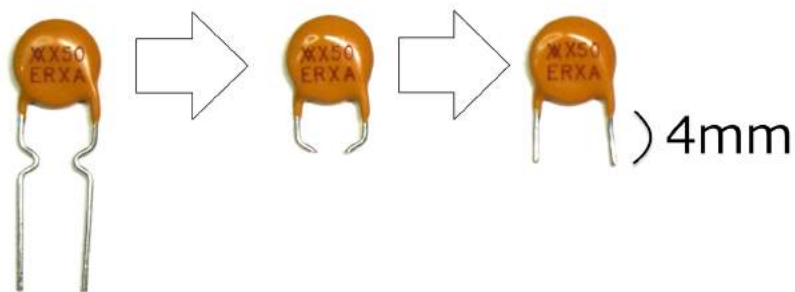
1. サーボモーター小

2. サーボモーター大

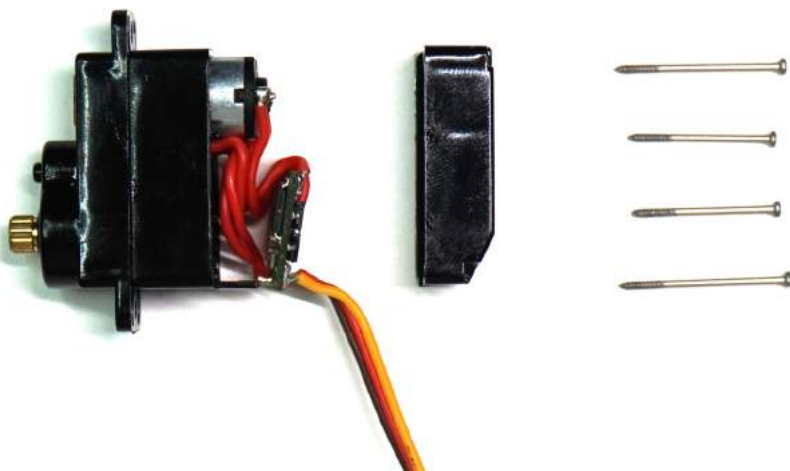
3. ポリスイッチ



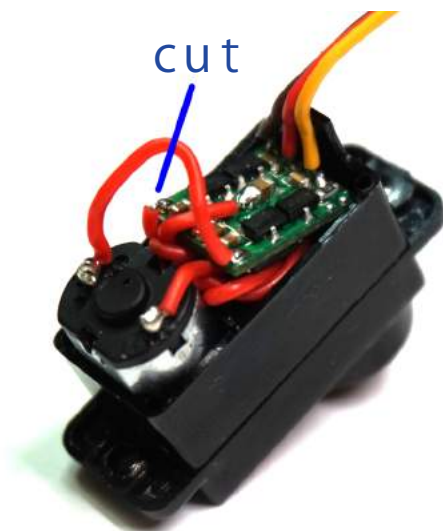
【14】始めに、ポリスイッチのリード線を4mm程残して切断し、ラジオペンチでまっすぐに伸ばします。この作業は3個分行います。



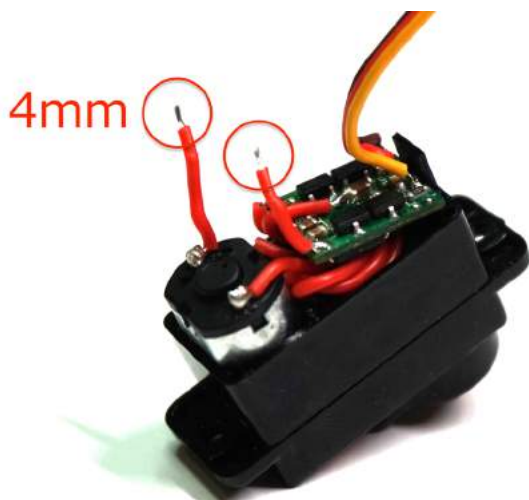
【15】サーボモーター小のねじを緩めてカバーを取り外します。側面にメーカーのシールが張ってあるので、それを剥がすか、部品の繋ぎ目でカットしてください。



【16】中にあるDCモーターにはんだ付けされている2本のケーブルのうちいずれか一本を写真の青線の部分で切断します。



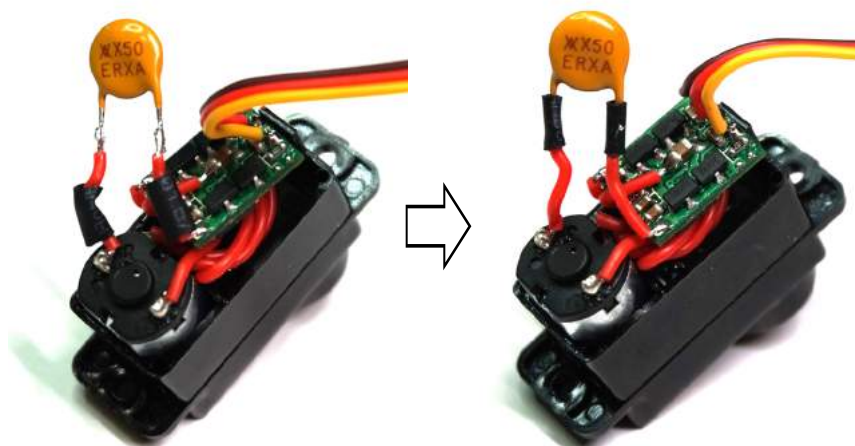
【17】ワイヤーストリッパーなどを用いて切断したケーブルの被覆を4mm程剥き、芯線はねじっておきます。



【18】切ったケーブルにポリスイッチをはんだ付けしてビニールテープで覆います。金属部分を露出させると危険なのでしっかり金属部分全体を覆って下さい。

※熱収縮チューブがあるなら熱収縮チューブを用いた方がより安全です。

※ポリスイッチに極性はありません。



【19】図のようにケーブルを畳んで、カバーを閉じ、ねじを締めます。この時、ケーブルを挟まないように気をつけてください。



【20】サーボモーター大のねじを緩めてカバーを取り外します。防水の為のゴムパッキンが出てきますが、最後カバーを閉める際には不要です。



【21】中にあるDCモーターにはんだ付けされている2本のケーブルのうちいずれか一本を写真のように切断し、被覆を4mm程ワイヤーストリッパーなどを用いて剥き、芯線をねじってまとめます。



【22】切ったケーブルにポリスイッチをはんだ付けしてビニールテープで覆います。**金属部分を露出させると危険なのでしっかり金属部分全体を覆って下さい。**

※熱収縮チューブがあるなら熱収縮チューブを用いた方がより安全です。

※ポリスイッチに極性はありません。



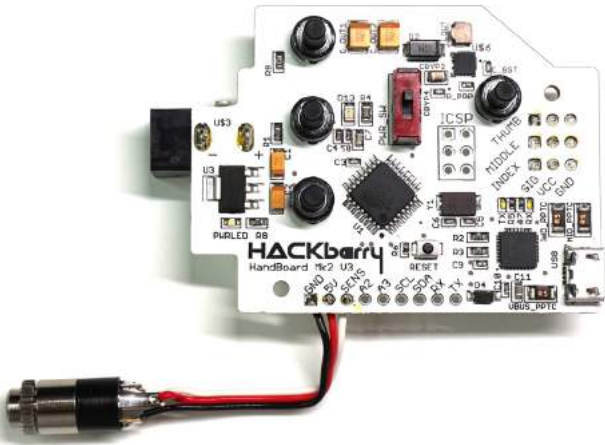
【23】カバーを閉じ、ねじを締めます。これでポリスイッチの挿入は完了です。



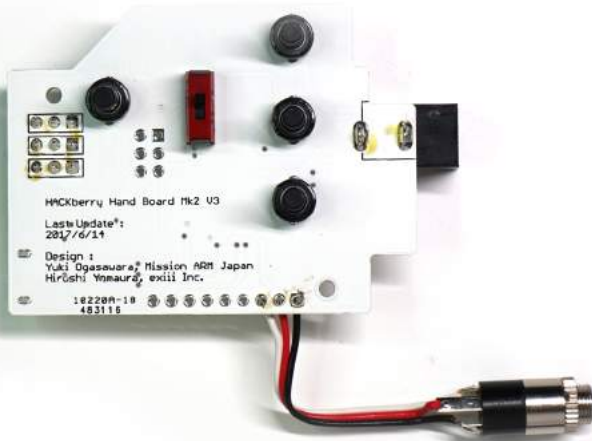
Mk2基板のはんだ付け

HACKberry hand board Mk2 は左右共用の制御基板ですが、**左右どちらに使用するかで裏表が逆になります**。右手用のはんだ付け方法の後に左手用のはんだ付け方法の記載があるので間違えないようにご注意ください。

Mk2基板完成イメージ(右手仕様)



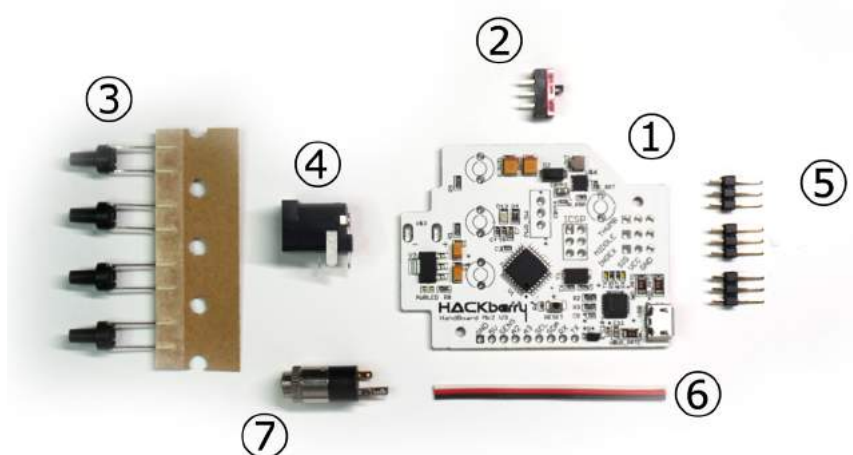
Mk2基板完成イメージ(左手仕様)



※基板のバージョンによって上の写真と外観が若干異なる場合があります。

—使用部品—

- | | | |
|-----------------------------|-------------|-------------|
| 1. HACKberry hand board Mk2 | 2. スライドスイッチ | 3. タクトスイッチ |
| 4. DCジャック | 5. ピンヘッダ | 6. フラットケーブル |
| 7. ステレオジャック | | |



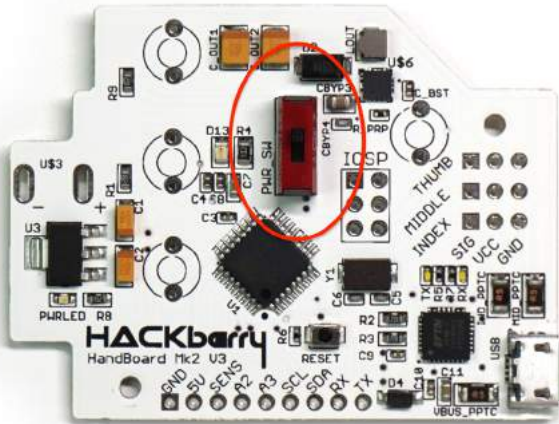
*はんだ付けの際は、電気素子(部品)をしっかりとおまて挿入し、浮いたり、斜めになったままはんだ付けをしないように注意してください。

*基板に乗っている小さな電気素子は静電気で破損する可能性があります。はんだ付けをする前に静電気除去パッド等を使って静電気を可能な限り除去してください。

*水に濡れた手で電気素子に触れると回路がショートして破損の原因になります。はんだ付け作業はしっかりと水気を拭き取ってから行ってください。

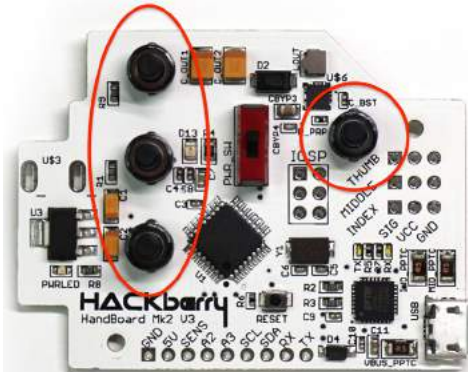
右手に用いる場合のはんだ付け

【1】スライドスイッチを基板のオモテ側にはんだ付けします。基板はHACKberryのロゴが印刷されている面がオモテです。またスライドスイッチに極性はありません。



【2】タクトスイッチを基板のオモテ側にはんだ付けします。ウラ側に突き出て余ったリード線は根本でカットします。目安としては基板表面から2mm以内でカットするようにします。

オモテ側



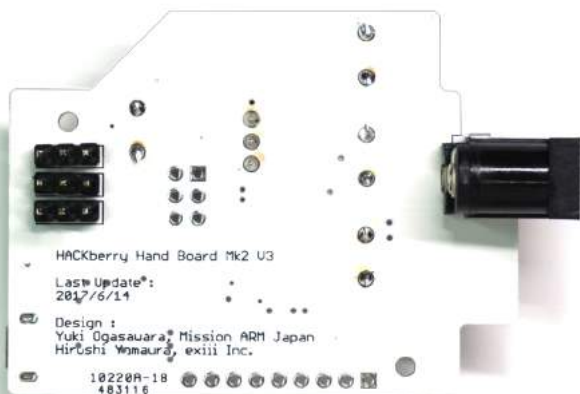
ウラ側



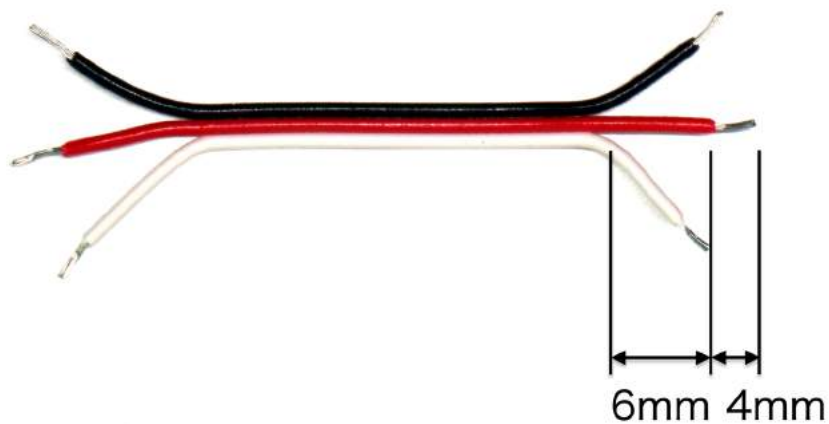
【3】DCジャックを図のようにはんだ付けします。



【4】ピンヘッダは**金属部分の短い方**を基板に挿してはんだ付けします。**斜めに傾きやすい**のでに気をつけてください。



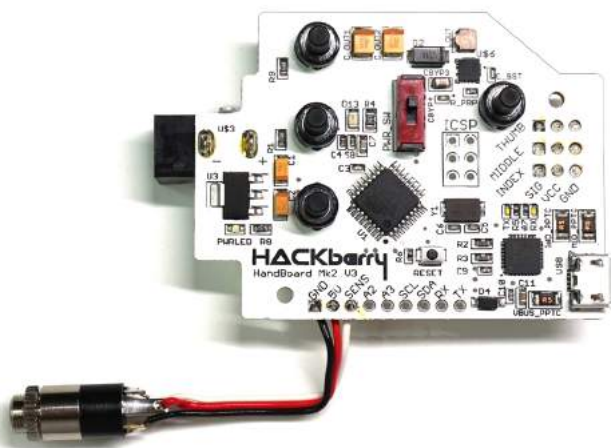
【5】フラットケーブルの両端をそれぞれ10mm程度割り、更に皮膜をそれぞれ4mm程度ワイヤーストリッパーなどを用いて剥きます。芯線は手でねじってまとめます。



【6】ステレオジャックの端子にフラットケーブルをはんだ付けします。この時、**右上の端子に白、下の端子に赤、左上の端子に黒のケーブル**を間違えないようにはんだ付けします。

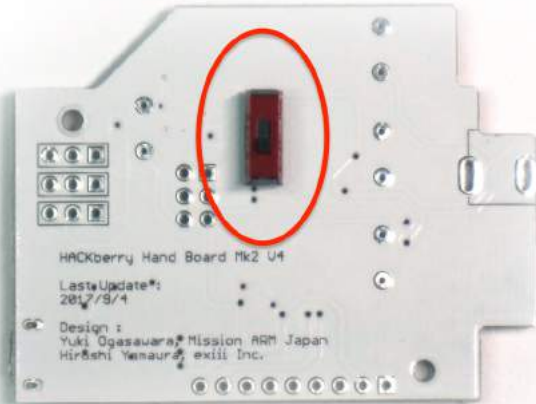


【7】フラットケーブルの反対側を基板にはんだ付けします。白いケーブルがSENS、赤が5V、黒がGNDに対応しています。これでHACKberry hand board Mk2のはんだ付けは終了です。



左手に用いる場合はんだ付け

【1】スライドスイッチを基板のウラ側にはんだ付けします。基板はHACKberryのロゴが印刷されていない面がウラ面です。またスライドスイッチに極性(向き)はありません。

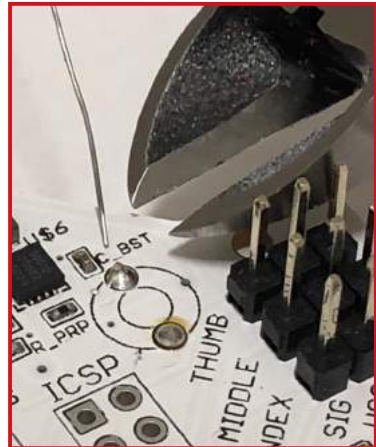


【2】タクトスイッチを基板のオモテ側にはんだ付けします。ウラ側に突き出て余ったリード線は根本でカットします。目安としては基板表面から2mm以内でカットするようにします。

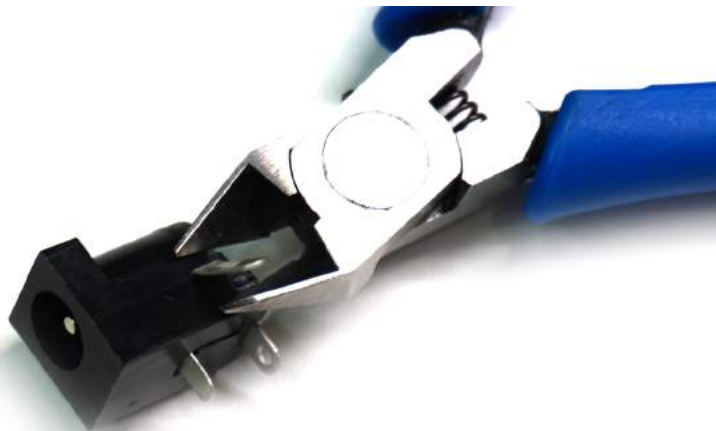
ウラ側



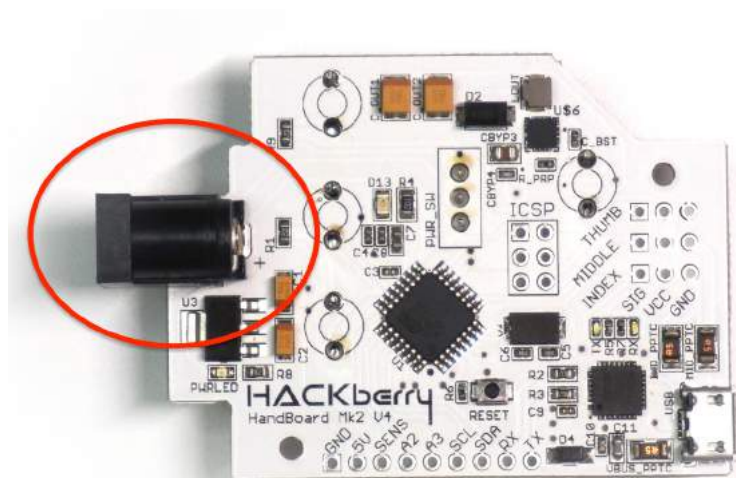
オモテ側



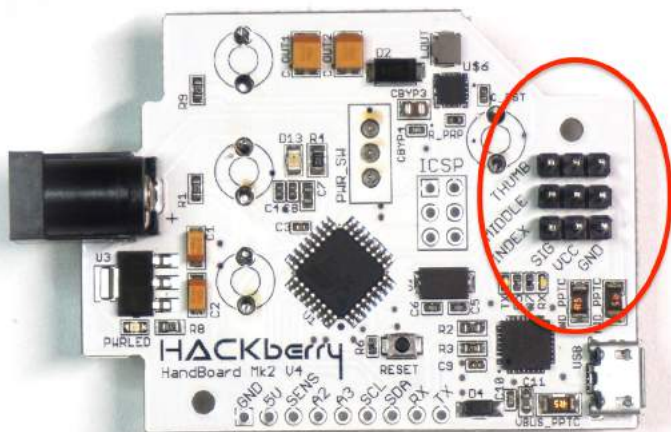
【3】次にオモテ面のはんだ付けを行います。DCジャックの横から出ている金属端子をニッパーでカットします。



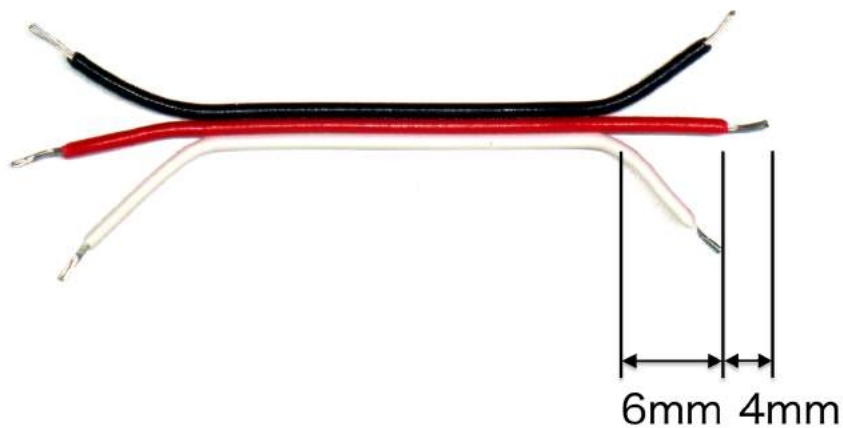
【4】DCジャックを図のようにはんだ付けします。



【5】ピンヘッダは短い方を基板に挿してはんだ付けします。斜めに傾き易いので気をつけてください。



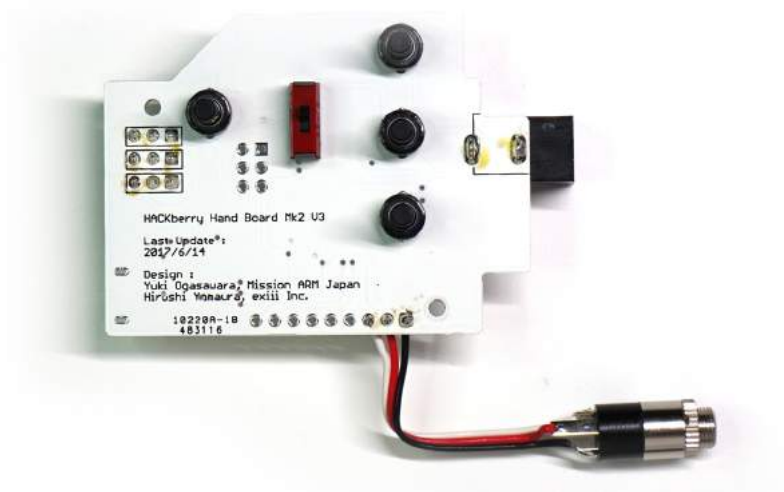
【6】フラットケーブルの両端をそれぞれ10mm程度割き、更に皮膜をそれぞれ4mm程度ワイヤーストリッパーなどを用いて剥きます。芯線は手でねじってまとめます。



【7】ステレオジャックの端子にフラットケーブルをはんだ付けします。この時、**右上の端子に白、下の端子に赤、左上の端子に黒のケーブル**を間違えないようにはんだ付けします。



【8】フラットケーブルの反対側を基板にはんだ付けします。**白いケーブルがSENS、赤が5V、黒がGND**に対応しています。これでHACKberry hand board Mk2基板のはんだ付けは終了です。



Assembling

05 プログラムのアップロード

基板の種類と左右の場合でのプログラム変更方法

ここでは、Windowsを用いたプログラムのアップロード方法について説明します。

【1】Arduinoの公式ホームページ <https://www.arduino.cc/en/Main/Software#>
にアクセスし最新バージョンのArduino IDEソフトウェアをダウンロードします。

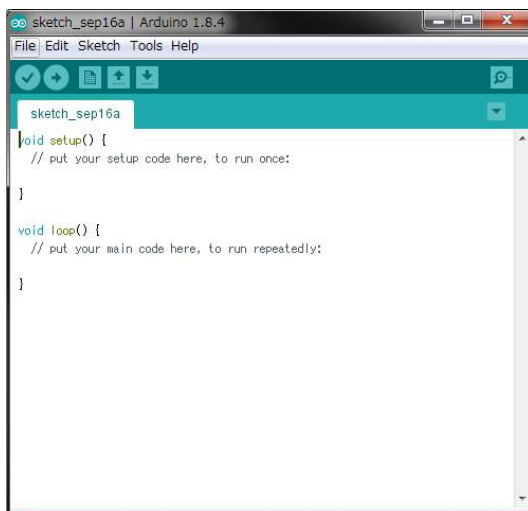
*本書執筆当時とはホームページのデザインやバージョンが異なる場合があります。



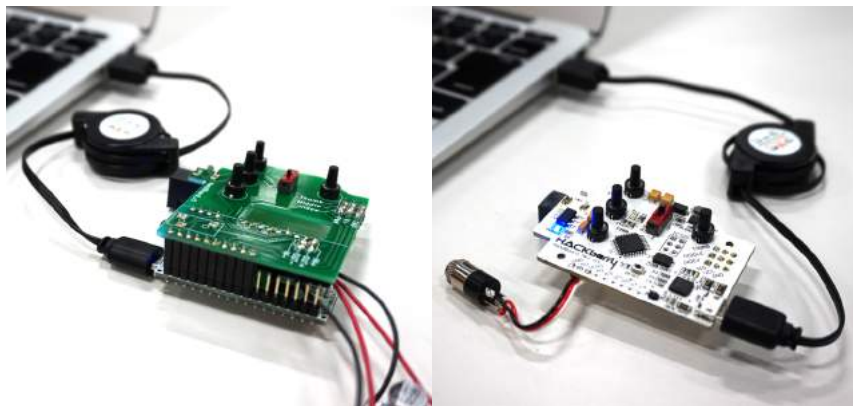
Download the Arduino IDE

A screenshot of the Arduino IDE download page. The main content area has a light blue background. On the left is the Arduino logo, a teal circle with a white infinity symbol containing a minus sign on the left and a plus sign on the right. To the right of the logo, the text reads: **ARDUINO 1.8.4**, followed by a paragraph describing it as open-source software that runs on Windows, Mac OS X, and Linux. Below this is a teal sidebar with white text listing download options: **Windows installer**, **Windows ZIP file for non admin install**, **Windows app** (with a 'Get it' button), **Mac OS X 10.7 Lion or newer**, **Linux 32 bits**, **Linux 64 bits**, **Linux ARM**, **Release Notes**, **Source Code**, and **Checksums (sha512)**. Below the main content area are two more sections: **ARDUINO SOFTWARE HOURLY BUILDS** with a 'LAST UPDATE 29 August 2017 11:58 GMT' badge, and **ARDUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES**.

【2】Arduinoのインストールファイルを実行し、指示に従ってパソコンにArduinoをインストールします。インストールが正しく完了したらArduino IDEを起動します。

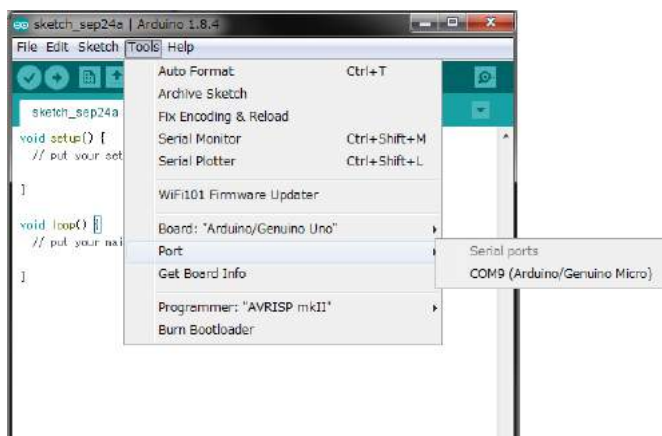


【3】Mk1基板の場合はArduino MicroとPCを、Mk2基板の場合はMk2基板とPCをマイクロUSBケーブルで接続します。



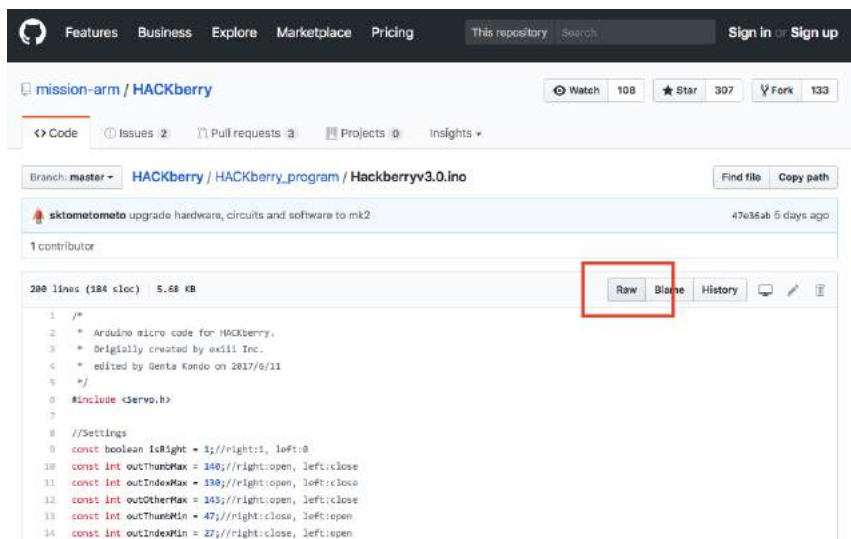
【4】自動でドライバーのインストールが始まります。インストールが完了するとArduino IDEのツールタブ内のポート欄に、新たにCOMポートが表示されるのでそれを選択します。

※表示されない場合は一度Arduino MicroまたはMk2基板からマイクロUSBケーブルを抜き、再度接続します。

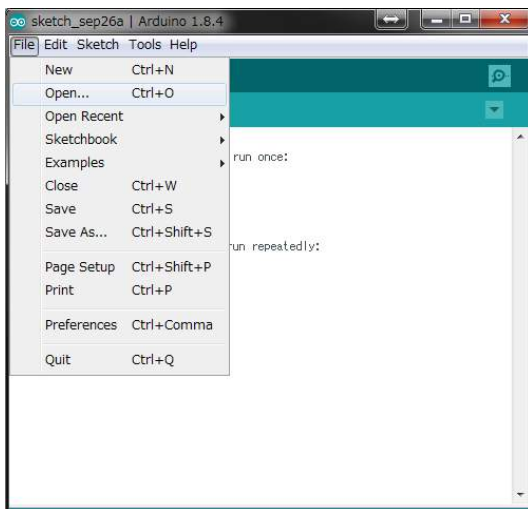


【5】GithubのHACKberryリポジトリの [HACKberry/HACKberry_program/Hackberryv3.0.ino](#) にアクセスし、手元にHACKberry hand board Mk1があれば“Hackberryv3.0”を、HACKberry hand board Mk2があれば“HackberryMk2v1.0”を開いて、「RAW」ボタンを右クリックし、「リンク先に名前をつけて保存する」を選びます。

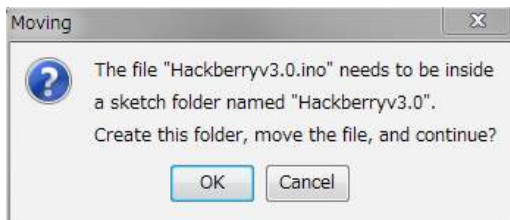
※プログラムのバージョンは本書執筆当時と変わっている場合があります。



【6】Arduino IDEの「ファイル → 開く」から先ほどダウンロードしたファイルを選択して開きます。

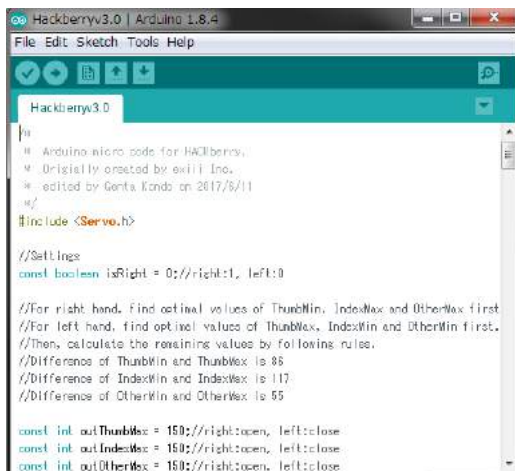


【7】途中で以下のようなメッセージが出てくる場合がありますがOKをクリックして、進めます。



【8】次に「ツール → マイコンボード」から該当するマイコンボードを選択します。最後に画面左上にある「▶」ボタンを押してプログラムをアップロードします。

※マイコンボードは、Mk1基板を使用している場合はArduino Micro、Mk2基板を使用している場合はArduino Nanoを選択します。



```
Hackberry3.0 | Arduino 1.8.4
File Edit Sketch Tools Help
Hackberry3.0
//
// Arduino-micro code for Hackberry.
// Originally created by exill Inc.
// edited by Genta Kondo on 2017/5/11
//
#include <Servo.h>

//Settings
const boolean isRight = 0;//right:1, left:0

//For right hand, find optimal values of ThumbMin, IndexMax and OtherMax first
//For left hand, find optimal values of ThumbMax, IndexMin and OtherMin first.
//Then, calculate the remainins values by followings rules.
//Difference of ThumbMin and ThumbMax is 86
//Difference of IndexMin and IndexMax is 117
//Difference of OtherMin and OtherMax is 55

const int outThumbMax = 150;//right:open, left:close
const int outIndexMax = 150;//right:open, left:close
const int outOtherMax = 150;//right:open, left:close
```


Assembling

06 バッテリーケース・ソケットの組み立て

バッテリーケースの組み立て

この節ではダミーソケットを含まないHACKberryに使用するバッテリーケースの組立方法を紹介합니다。もしダミーソケット付きのHACKberryを組み立てる場合は62ページから始めてください。



—使用部品—

1. C-B-01

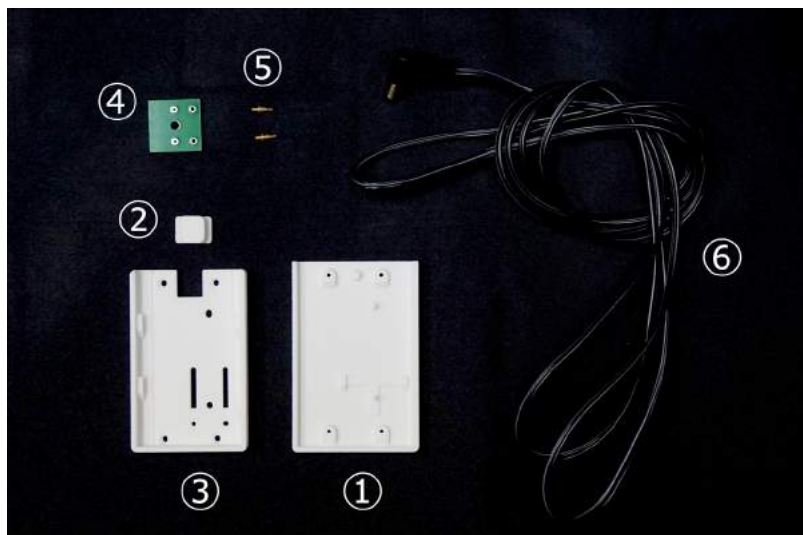
2. C-B-02

3. C-B-03

4. Battery board

5. バッテリープローブ

6. DCプラグ

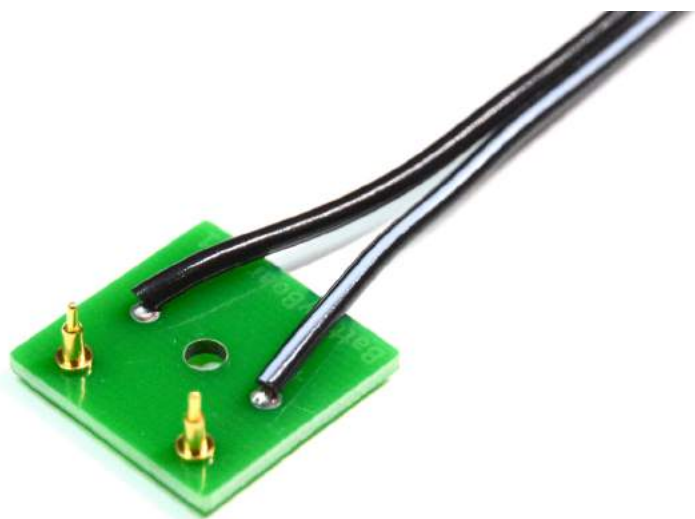
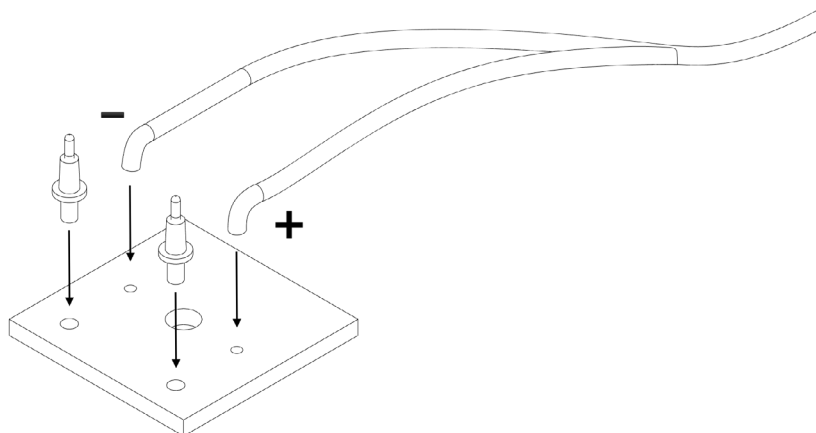


—使用汎用部品—

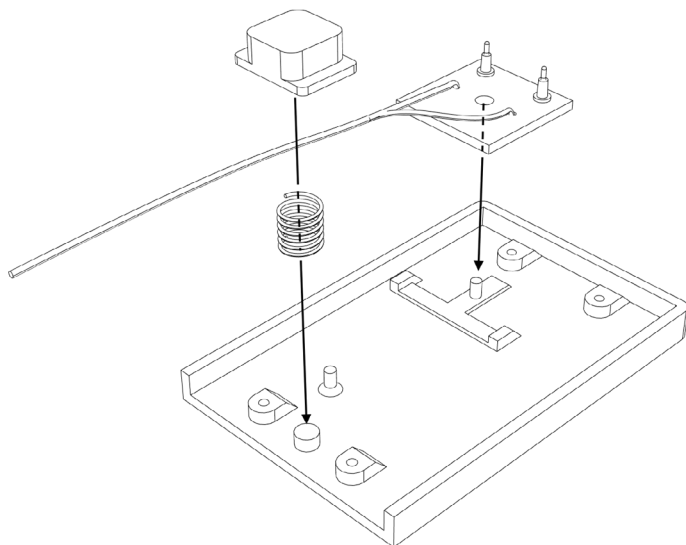
 x4

 x1

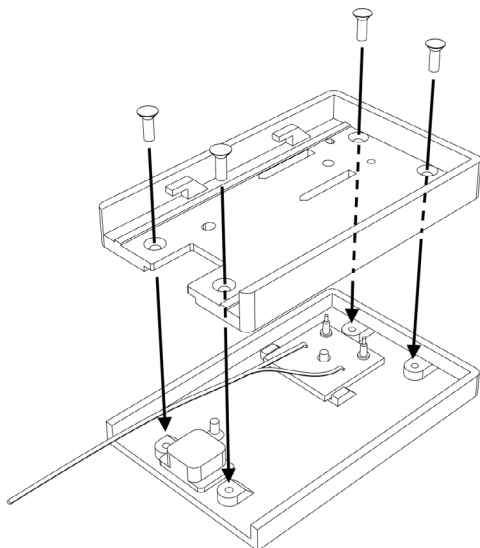
【1】バッテリーロープとDCプラグをBattery boardに下図のようにはんだ付けします。このとき、白いラインがある方のケーブルを+にはんだ付けするようにしてください。



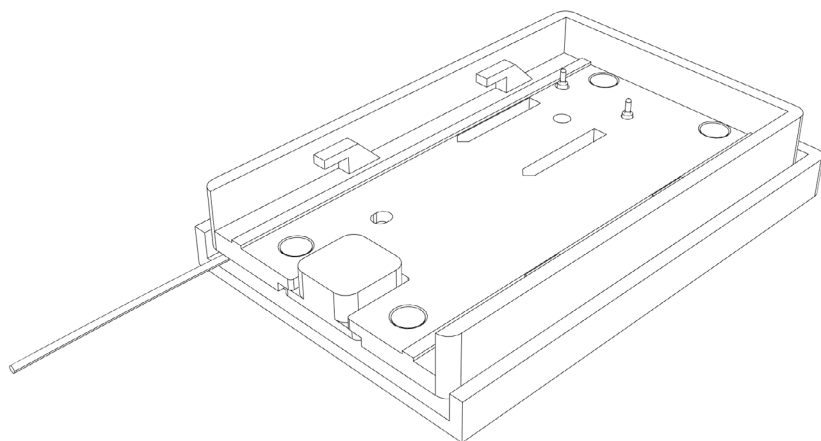
【2】コイルばねとC-B-02をC-B-01に下図のように置きます。



【3】C-B-03を4本のねじ(M2L6)で下図のように固定します。



【4】これでバッテリーケースの組み立ては完了です。

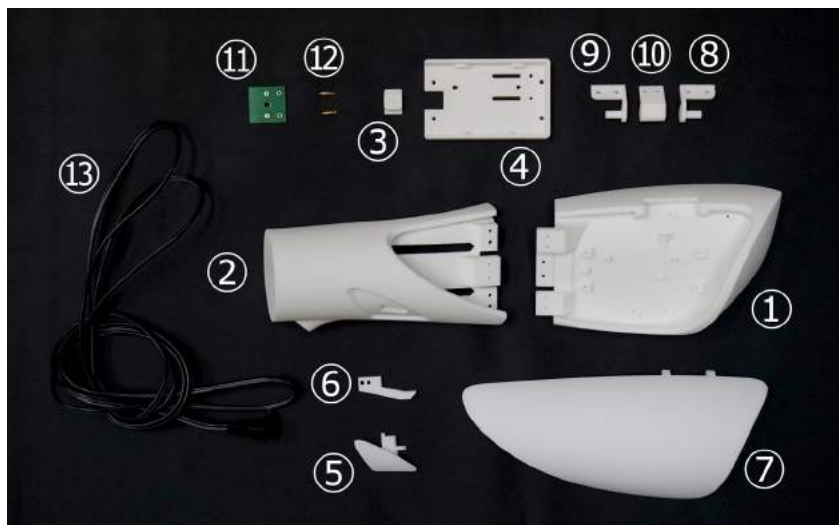


ダミーソケットの組み立て



—使用部品—

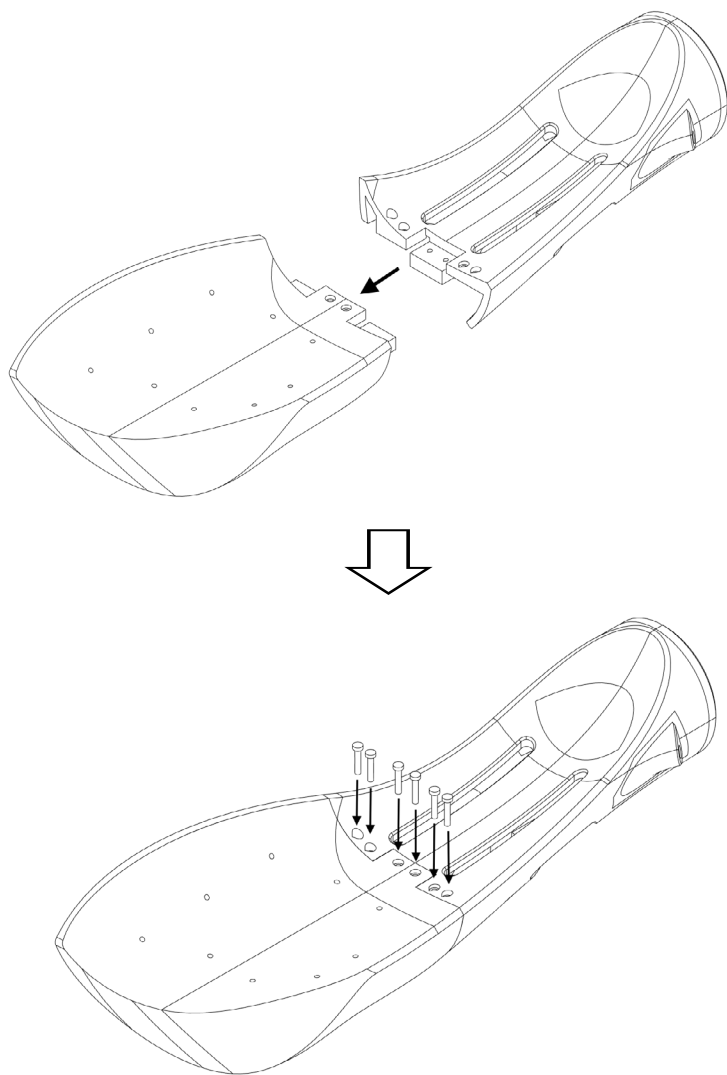
- | | | |
|-------------|-------------------|---------------|
| 1. R-SO-01 | 2. R-SO-02 | 3. R-SO-03 |
| 4. R-SO-04 | 5. R-SO-05 | 6. R-SO-06 |
| 7. R-SO-07 | 8. R-SO-08 | 9. R-SO-09 |
| 10. R-SO-10 | 11. Battery board | 12. バッテリーブローブ |
| 13. DCプラグ | | |



—使用汎用部品—



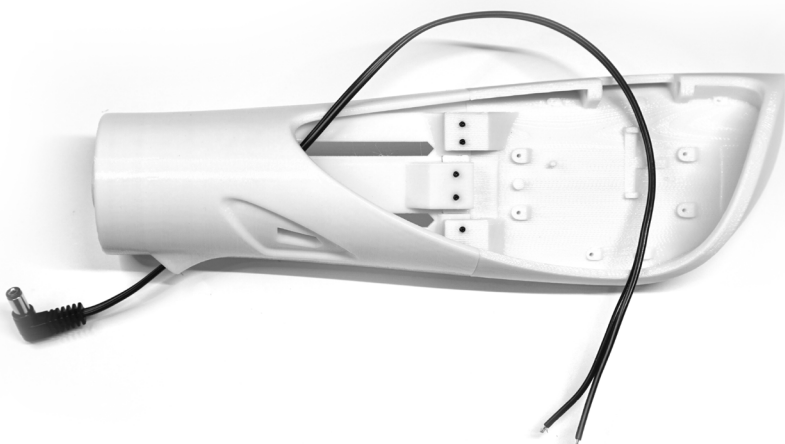
【1】R-SO-01とR-SO-02を6本のねじ(M2L10)で下図のように固定します。



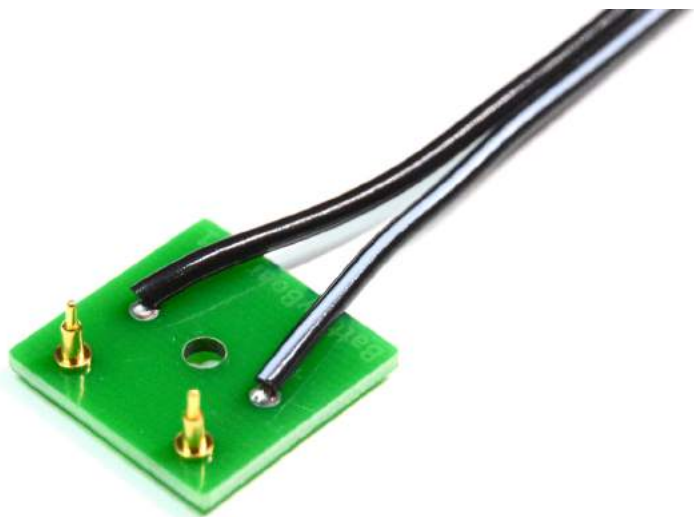
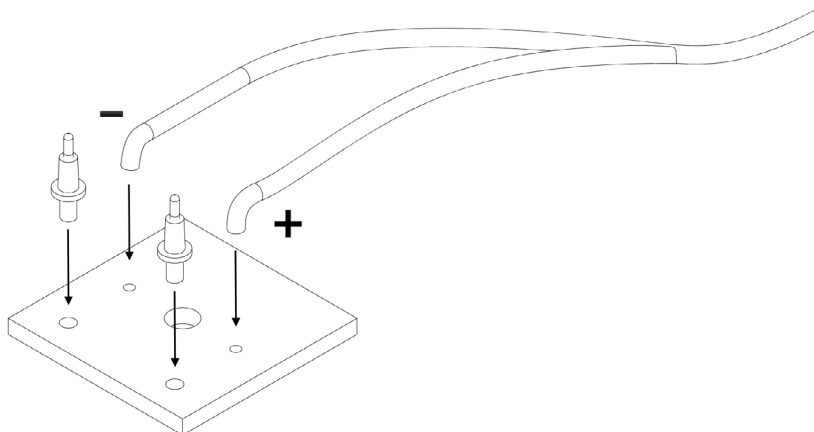
【2】DCプラグを33cmに切り、切断した端の被覆を4mm剥いた後に内部の芯線をねじってまとめます。もし手元にワイヤーストリッパーがない場合はニッパーで内部の芯線を切らないようにして被覆を剥きます。



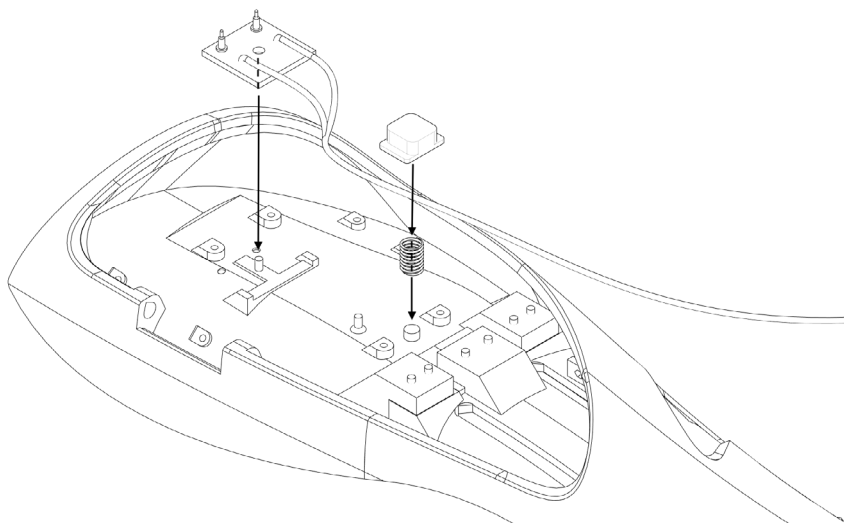
【3】その線をR-SO-02の穴に下図のように通します。



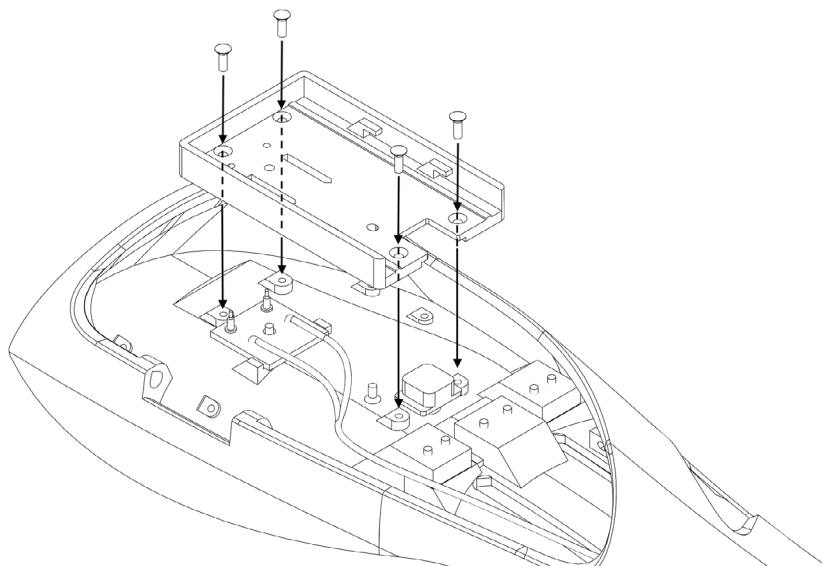
【4】バッテリーロープとDCプラグをBattery boardに画像のようにはんだ付けします。このとき、白いラインがある方のケーブルを+にはんだ付けするようにしてください。



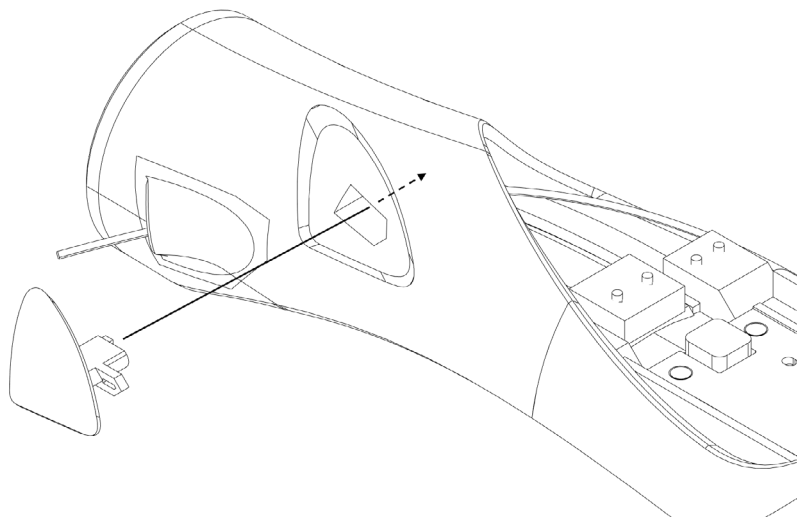
【5】コイルばねとR-SO-03を下図のように置きます。



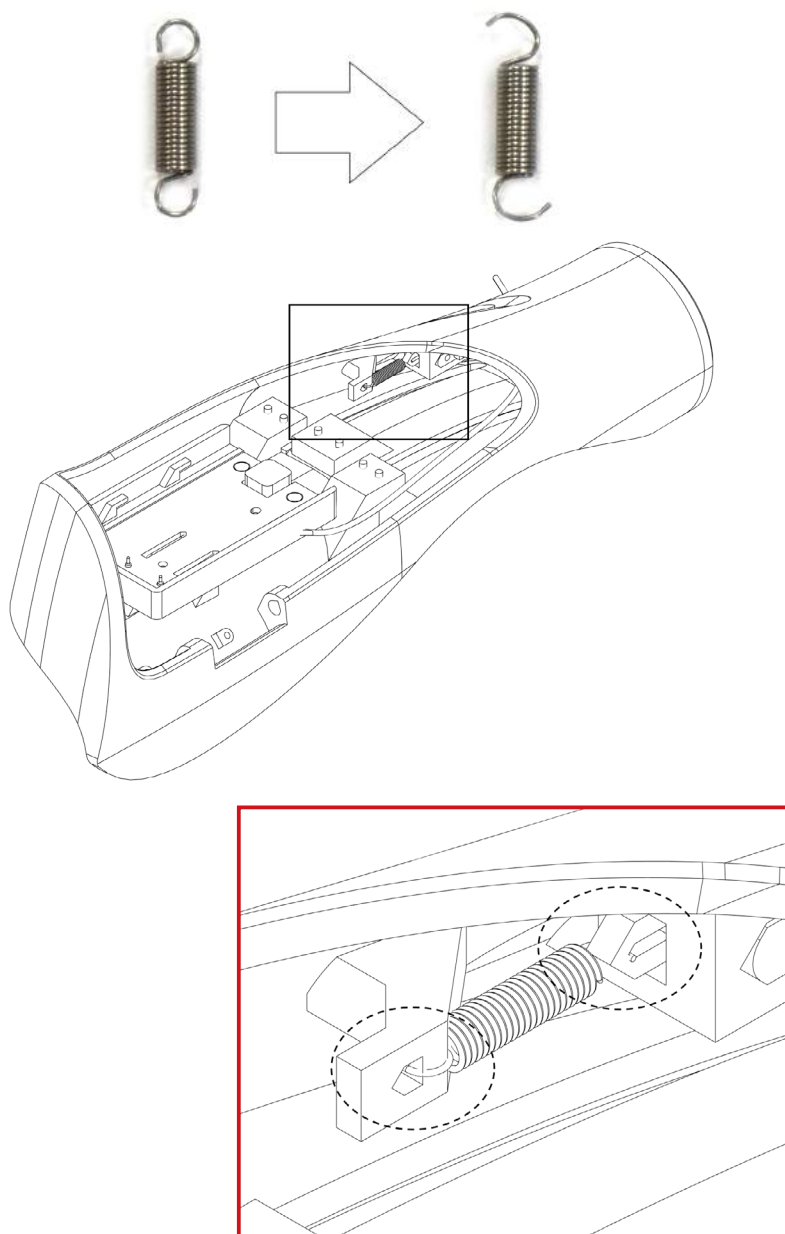
【6】R-SO-04を4本のねじ(M2L6)で下図のように固定します。



【7】図のようにR-SO-02にR-SO-05をはめます。

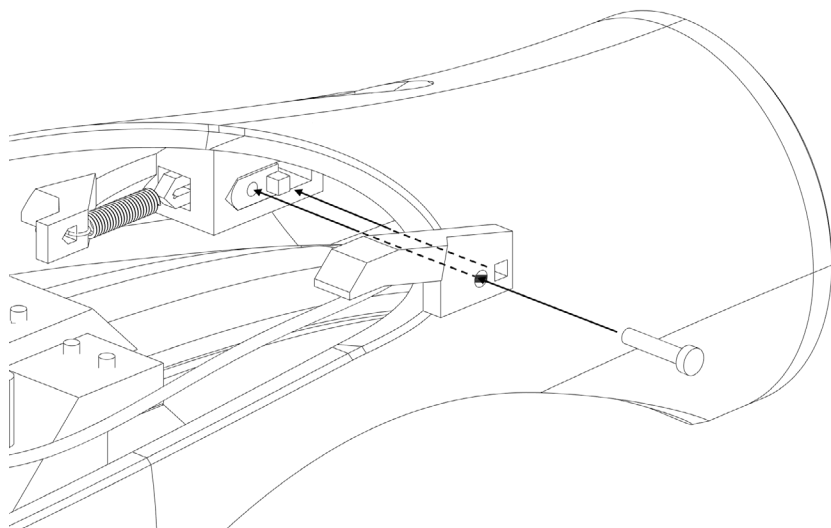


【8】引張ばねの両端を下図のように広げて取り付けます。

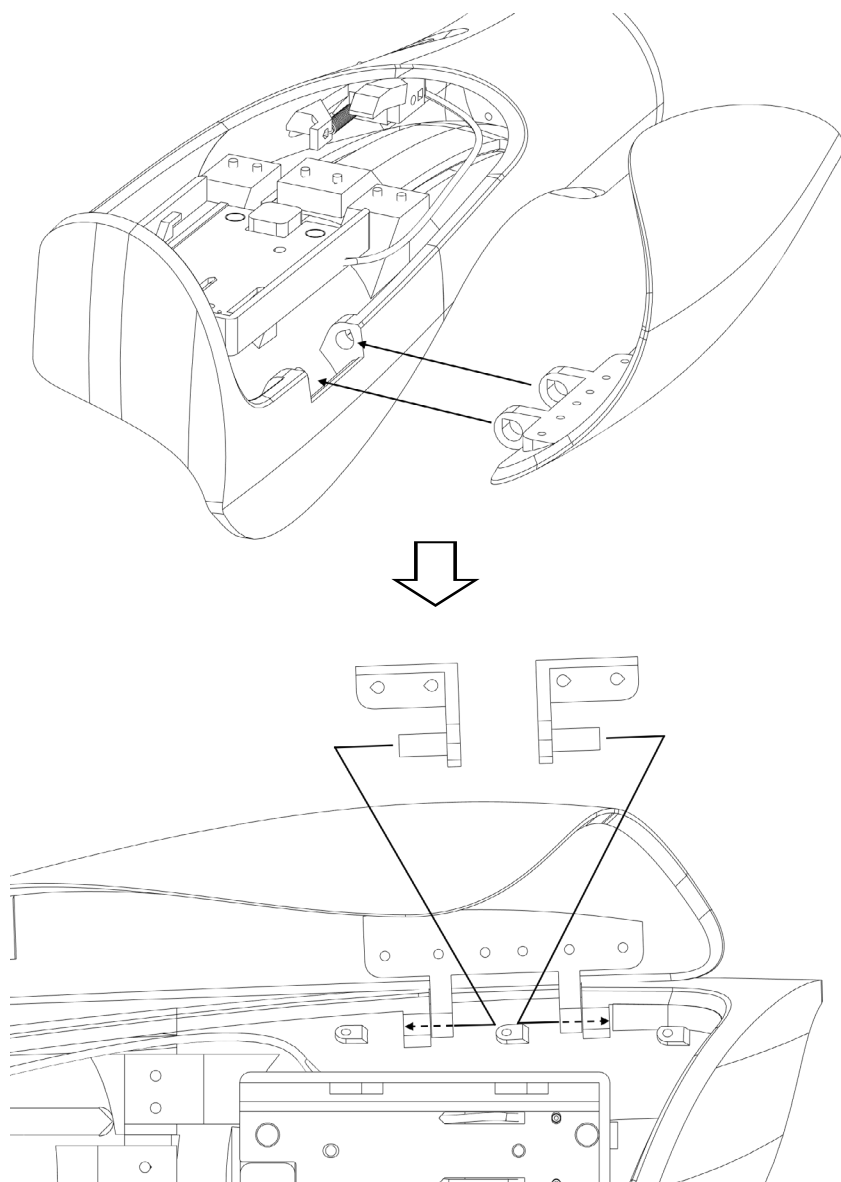


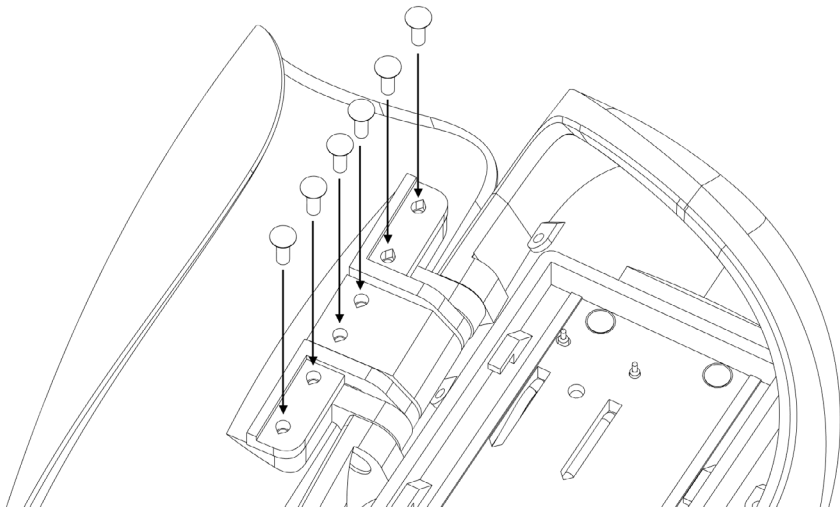
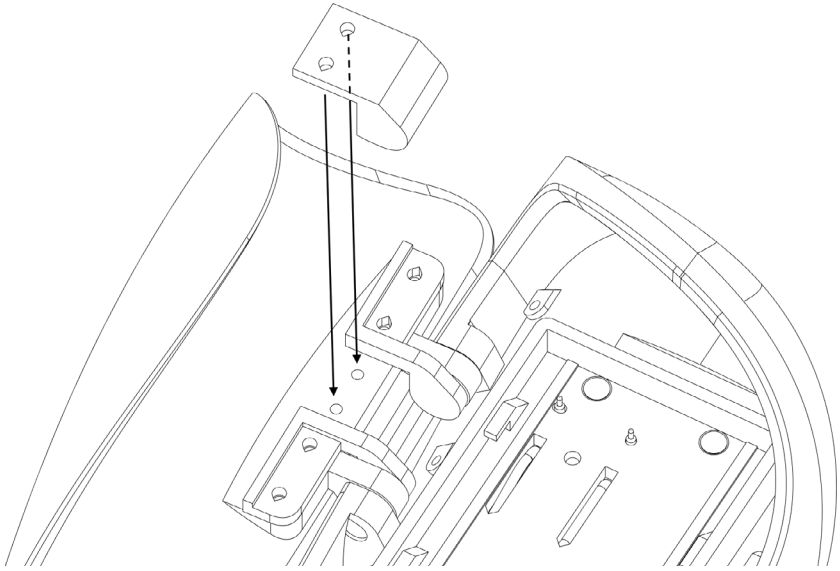
拡大図

【9】ねじ(M2L10)を用いてR-SO-06を図のように固定します。

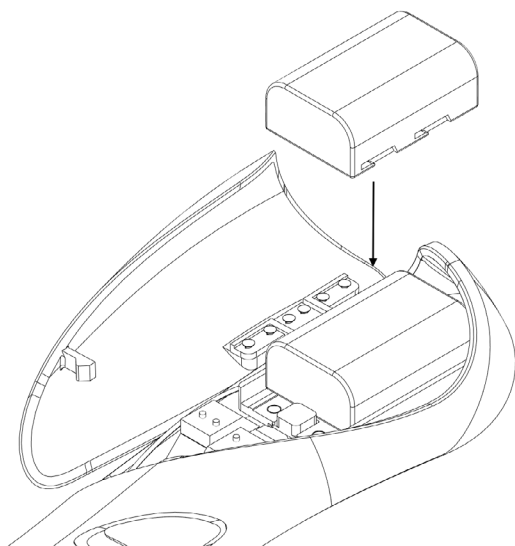


【10】R-SO-01とR-SO-07にR-SO-08とR-SO-09を通して6本のねじ(M2L6)で固定します。

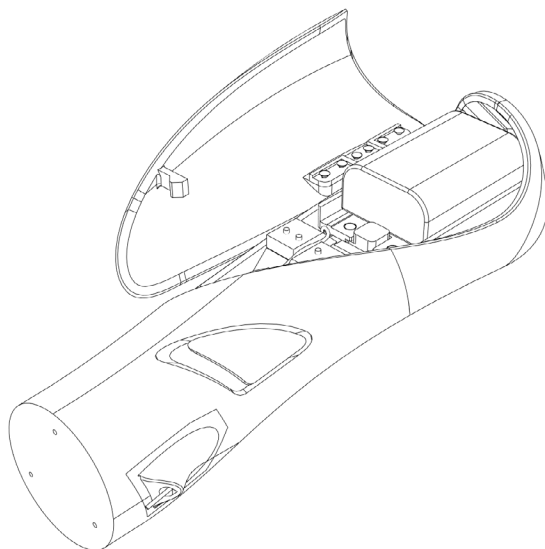




【11】バッテリーを取り付けます。



【12】これでソケットの組み立ては終了です。



Assembling

07 親指の組み立て



—使用部品—

1. R-T-01

2. R-T-02

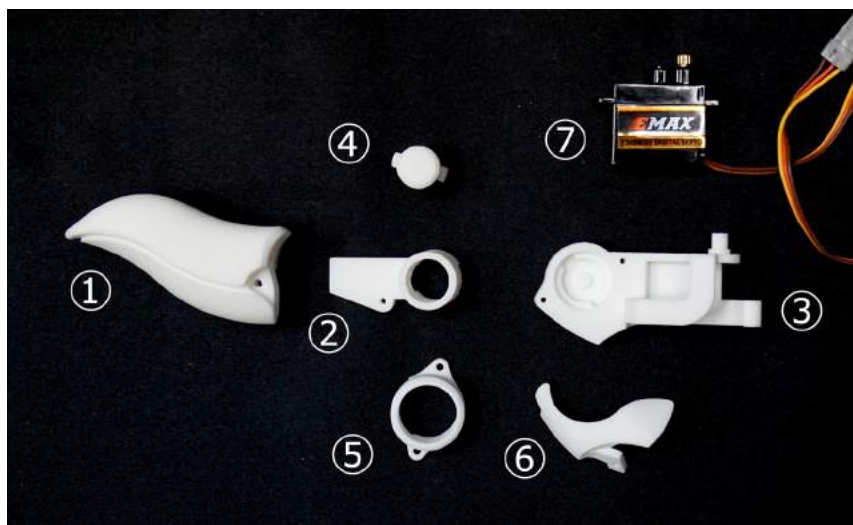
3. R-T-03

4. R-T-04

5. R-T-05

6. R-T-06

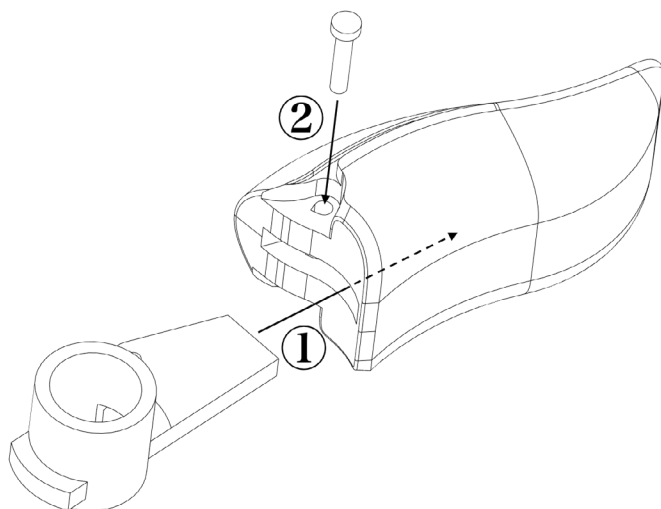
7. サーボモーター小



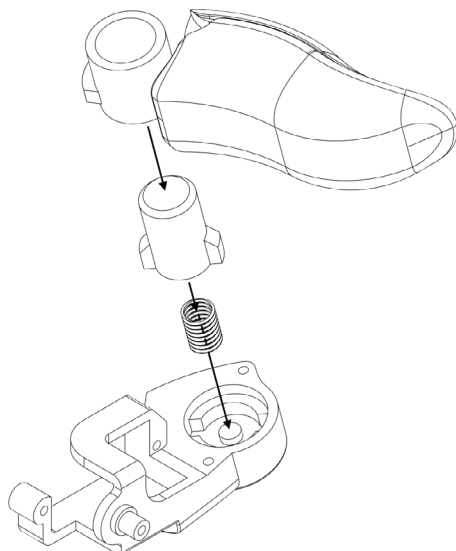
—使用汎用部品—



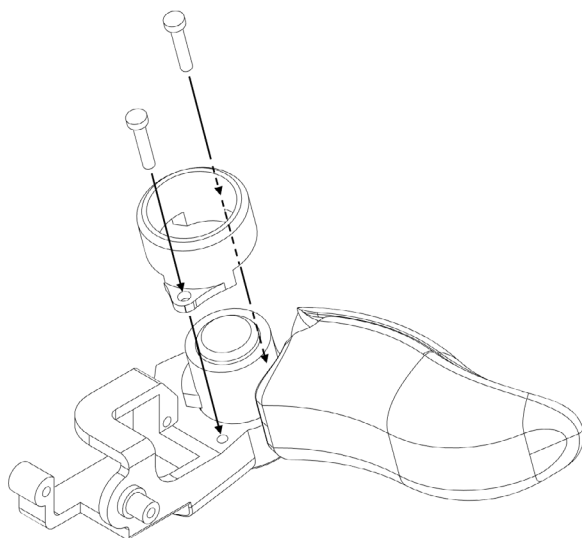
【1】R-T-01 に R-T-02 を通してねじ(M2L10)で止めます。



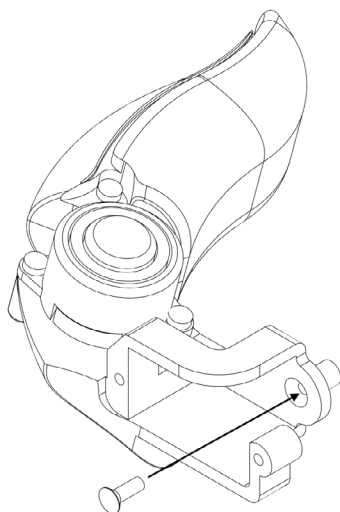
【2】コイルばねを下図のように R-T-03 に置き、R-T-04 を溝に合わせ、コイルばねに被せるように置きます。R-T-02 を通します。



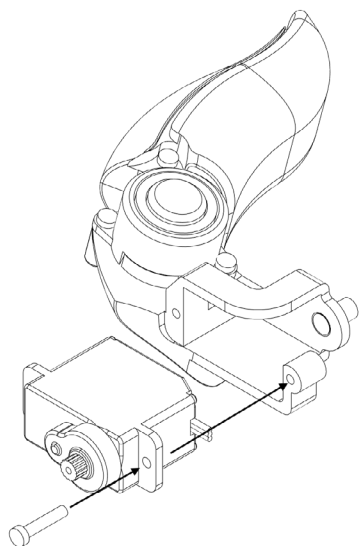
【3】R-T-05 を被せ、2本のねじ(M2L10)で固定します。



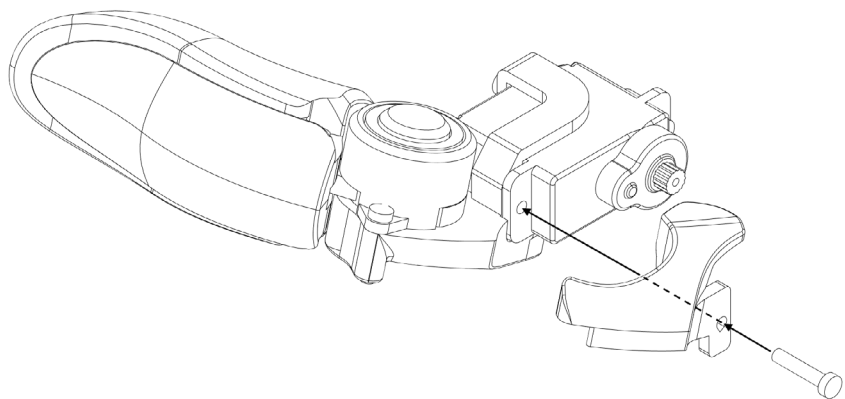
【4】下図のように R-T-03 にねじ(M2L6)を取り付けます。



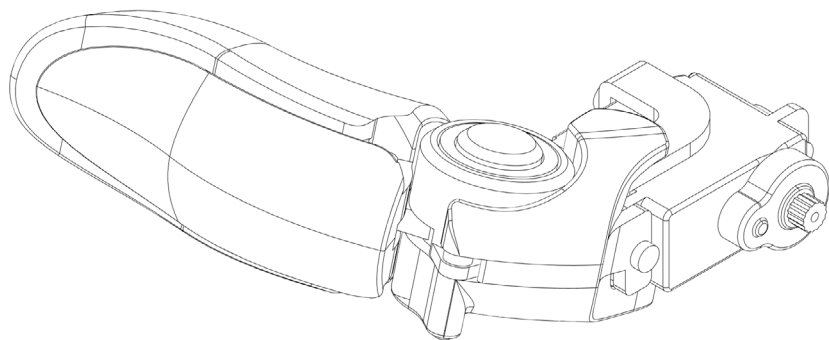
【5】サーボモーター小をねじ(M2L10)で固定します。



【6】その後 R-T-06 をねじ(M2L10)で固定します。



【7】これで親指の組み立ては終わりです。



Assembling

08 人差し指の組み立て



—使用部品—

1. R-I-01

2. R-I-02

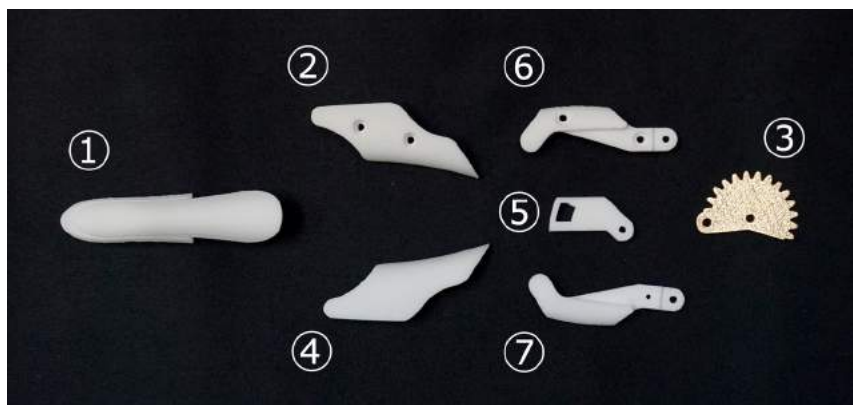
3. R-I-03

4. R-I-04

5. R-I-05

6. R-I-06

7. R-I-07



※写真のR-I-03は、強度を高めるため金属性の材料を使い3Dプリントしたものです。強度と耐久性は劣りますが、ほかの部品同様に樹脂製の材料をお使い頂くことも可能です。

—使用汎用部品—



x4

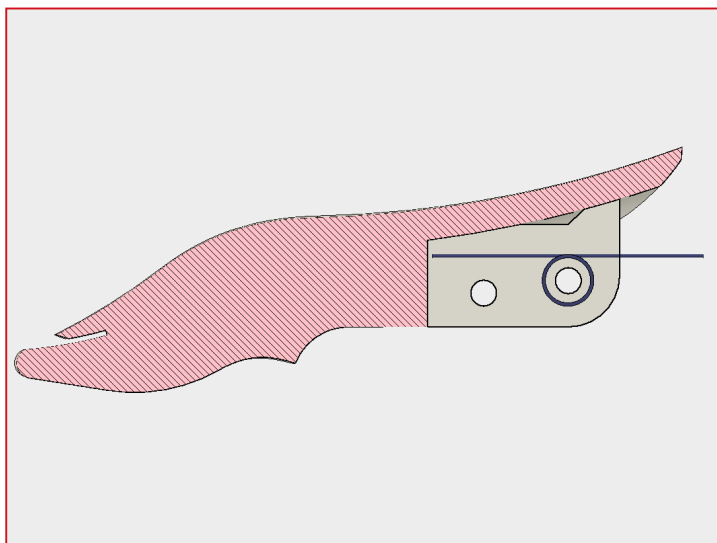
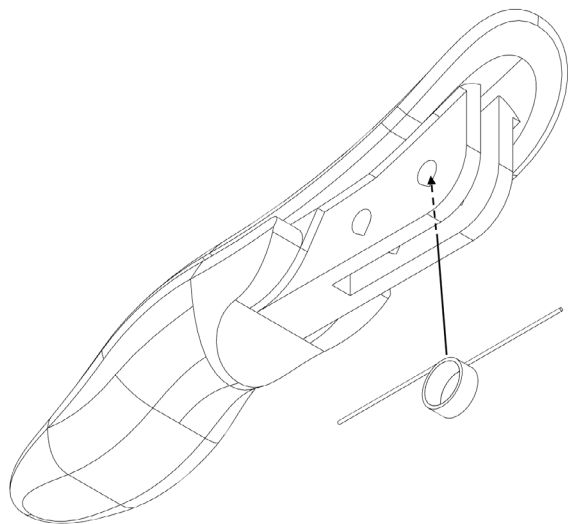


x1



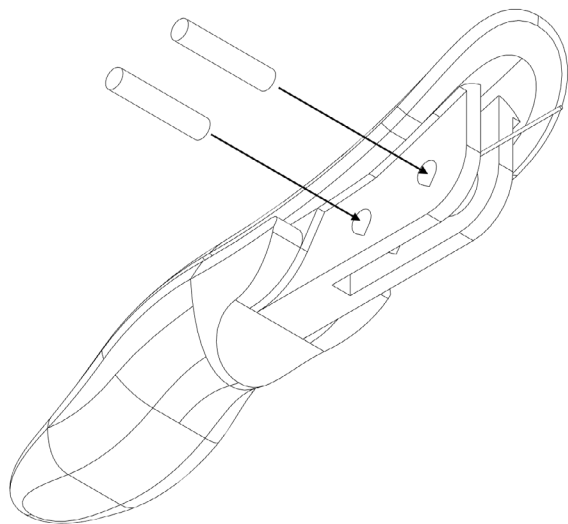
x4

【1】トーションばねを下図のように R-I-01 に入れ込みます。



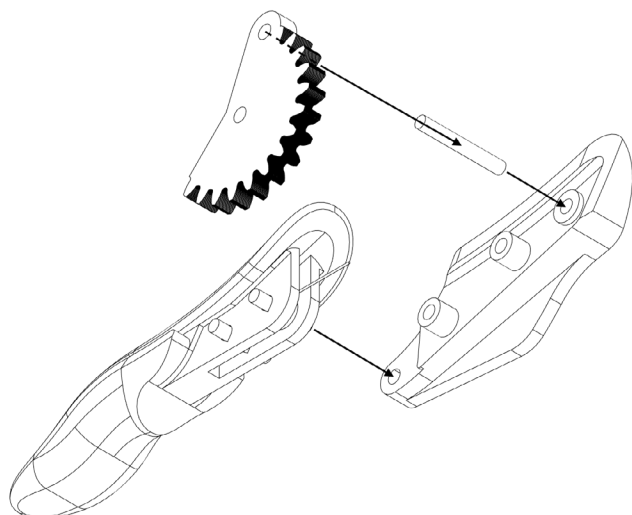
断面図

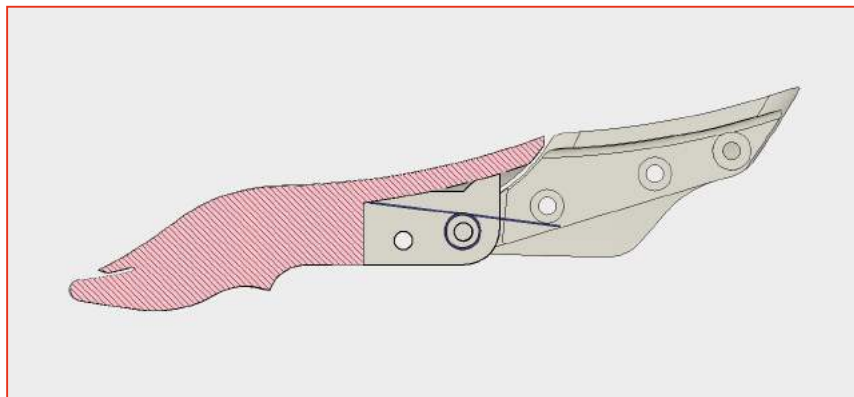
【2】そこにシャフト(10mm)を2本、下図のように差し込みます。この時、図右側のシャフトはトーショ
ンばねの円を通すよう注意して下さい。



【3】R-I-02 と R-I-03 を取り付けます。R-I-03は図のようにシャフト(10mm)を通して取り付けます。

※このときのトーションばねの位置に注意してください。



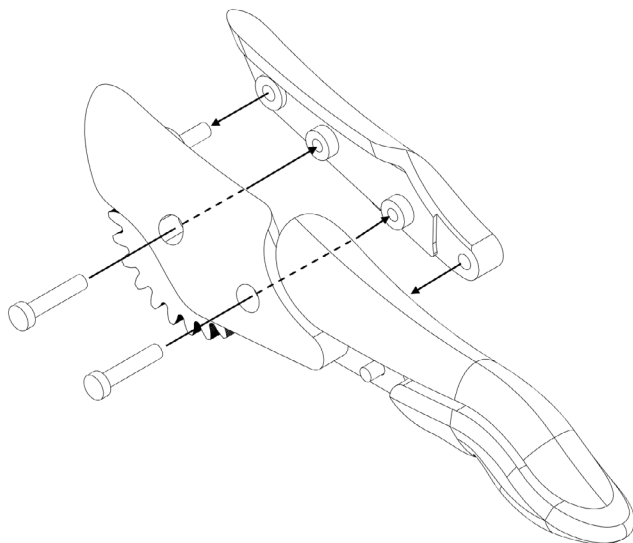


トーションばね位置 断面図

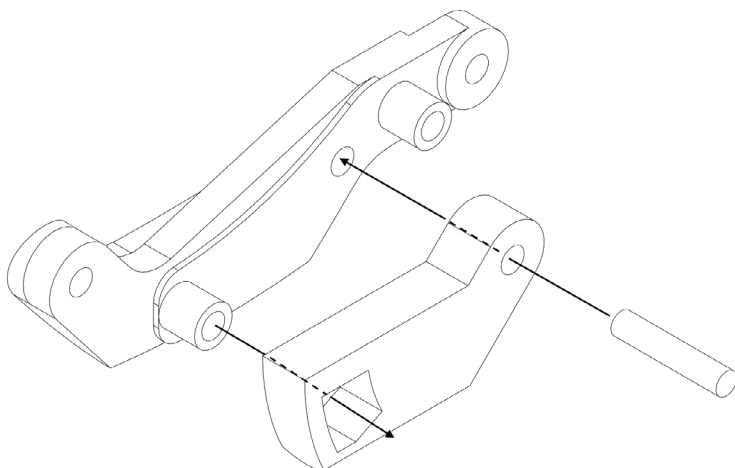


トーションばね位置 写真

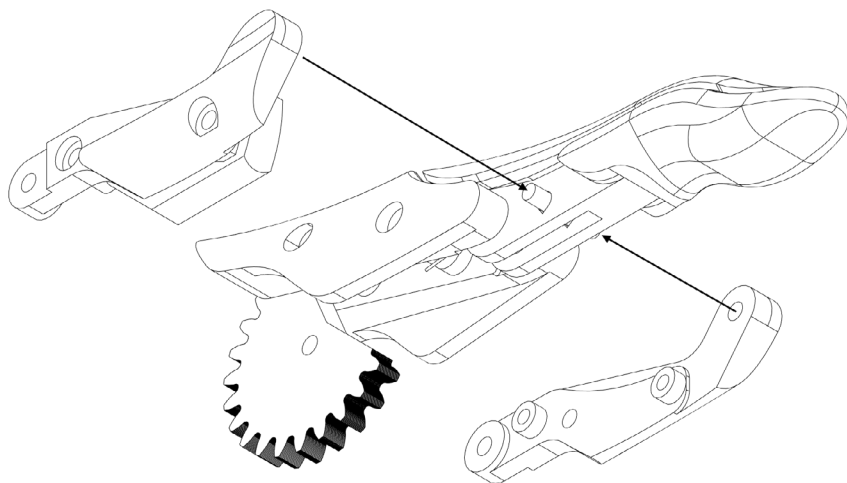
【4】R-I-04 を被せ、ねじ(M2L10)で2箇所固定します。



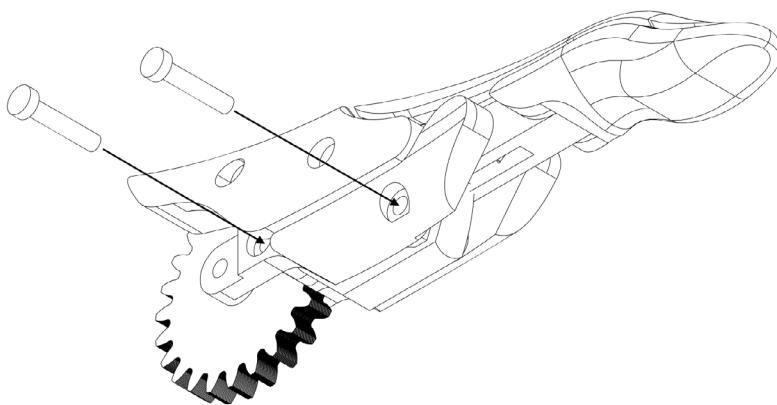
【5】シャフト(10mm)を R-I-05 に通した後それを R-I-06 に取り付けます。



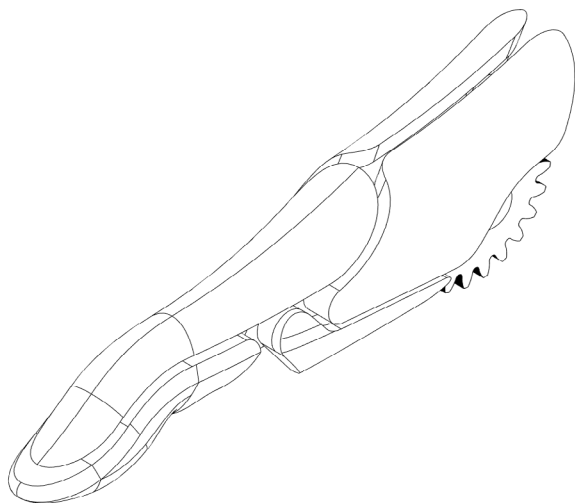
【6】R-I-07および【5】で組み立てた部品を使い、【4】で組み立てた部品を両側から挟み、組み合わせます。



【7】【6】で組み合わせた部品を図のようにねじ(M2L10)で2箇所固定します。



【8】これで人差し指は完成です。



Assembling

09 三指の組み立て

※中指、薬指、小指を組み立てます。これらは同じ部品を使います。



x3

—使用部品—

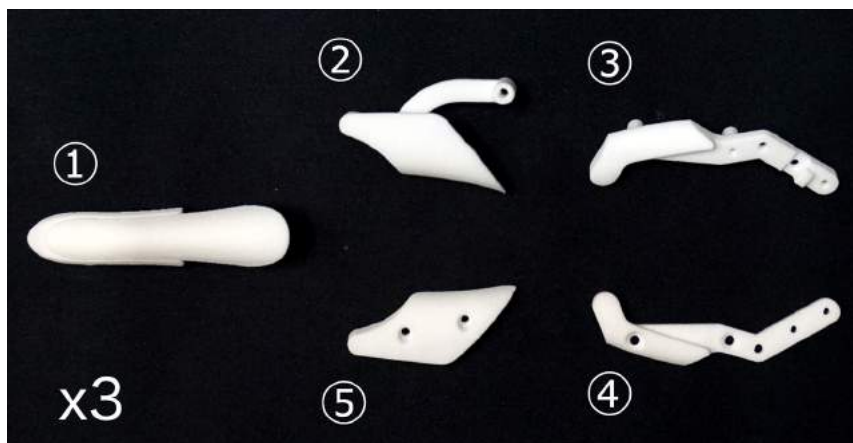
1. R-O-01

2. R-O-02

3. R-O-03

4. R-O-04

5. R-O-05



—使用汎用部品—

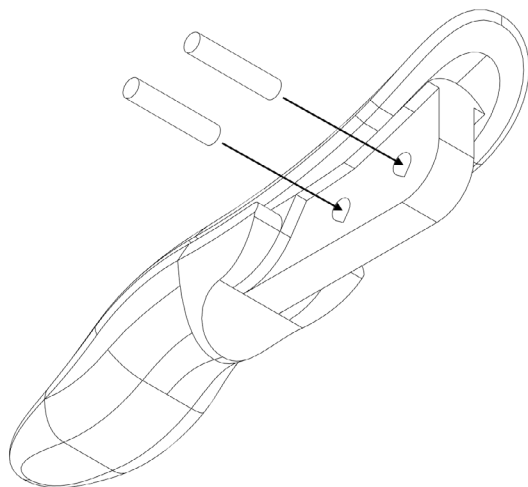


x12

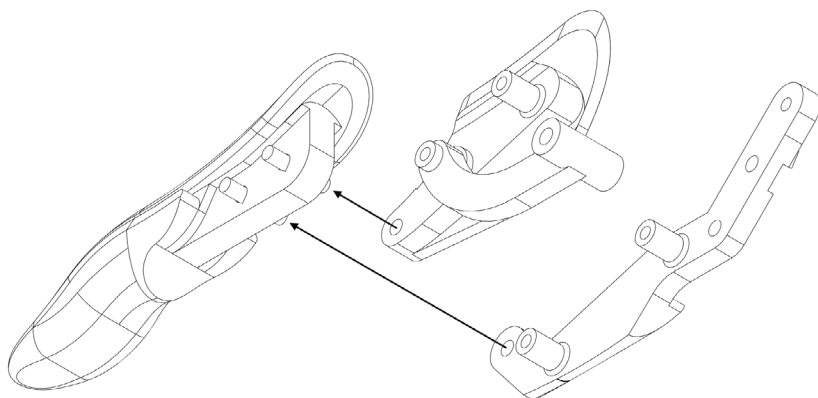


x6

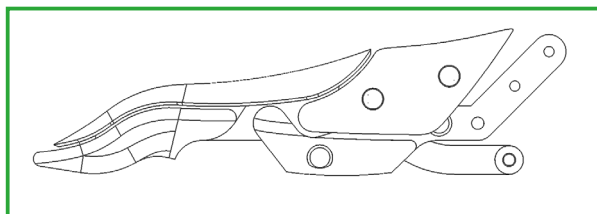
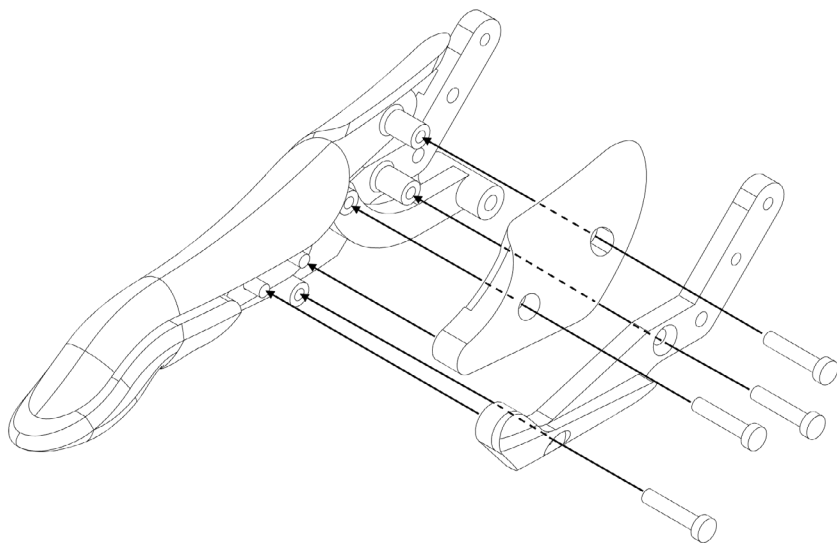
【1】 2本のシャフト(10mm)を R-O-01 に差し込みます。



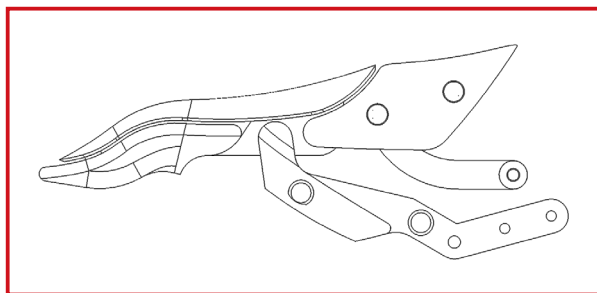
【2】 R-O-02 と R-O-03 を下図のように取り付けます。



【3】R-O-04 と R-O-05 を 4 本のねじ(M2L10)で固定します。

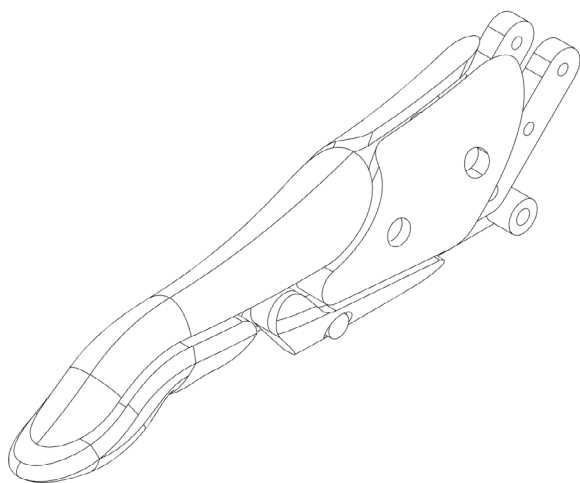


※良い例



※悪い例

【4】 同じものを3つ組み立てます。これで中指、薬指、小指は完成です。

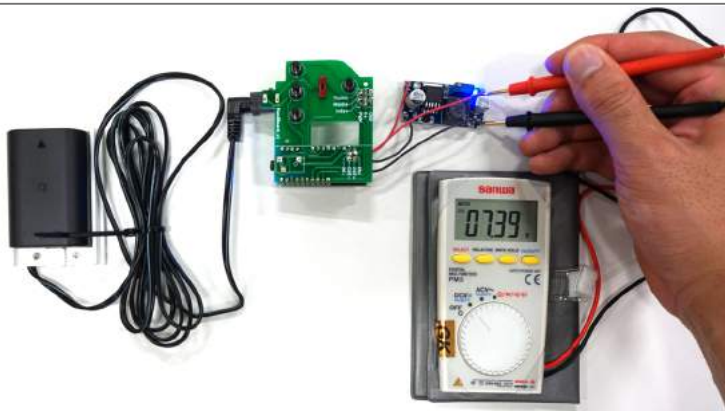


Assembling

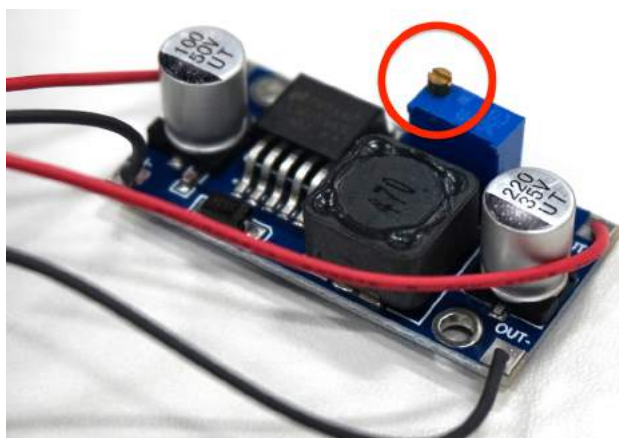
10 電圧の調整とサーボモーターの初期位置設定

ハンドを組み立てる前にサーボモーターの角度を設定します。HACKberry hand board Mk2を使う場合【1】【2】は必要ありません。

【1】HACKberry hand board Mk1にDCプラグを挿入し、スライドスイッチを上押し電源を入れます。DCDCコンバーターのOUT+とOUT-にテスターを当てて電圧を測ります。未調整の状態ではバッテリーの電圧がそのまま出力されています。



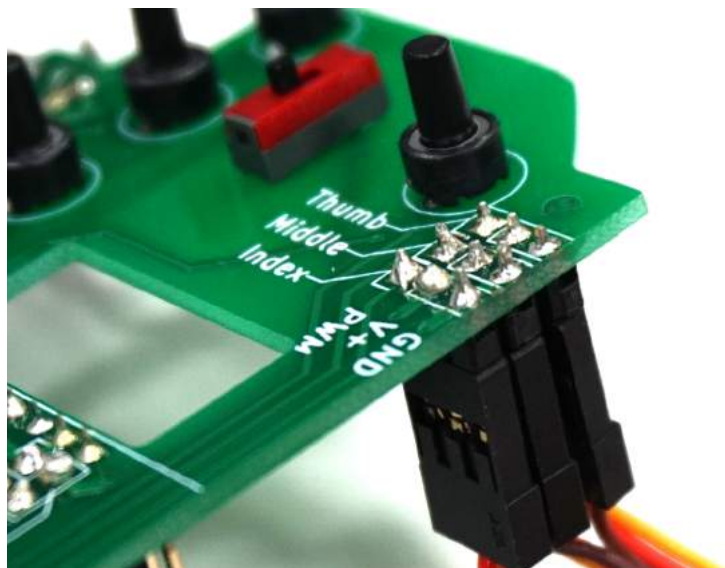
【2】この出力電圧を5Vに降圧させるために、DCDCコンバーター上の金色の部品をマイナスドライバーを用いて反時計回りに回していきます。回し始めて5, 6回転した辺りから徐々に電圧が低くなってきます。調整が終わったら一度電源を切ります。



【3】基板上のピンヘッダにサーボモーターのコネクタを指します。**黒(または茶)のケーブルが外側になるよう注意してください。**Indexと書かれたピンヘッダにはサーボモーター大、MiddleおよびThumbと書かれたピンヘッダにはサーボモーター小のコネクタを差します。

Mk1基板の場合、黒(または茶)がGND、赤がV+、白(または黄)がPWMにつながります。

Mk2基板の場合、黒(または茶)がGND、赤がVCC、白(または黄)がSIGにつながります。

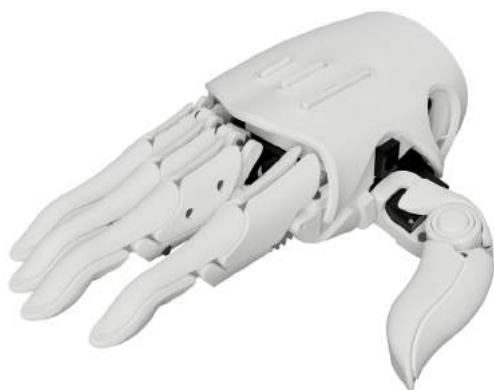


接続後、基板の電源を投入してモーターの角度が変わったら調整は終了です。

サーボの種類が同じ**MiddleとThumbのサーボモーターには印を付けておき**、後ほど再接続する際に間違えないようにします。

Assembling

11 掌の組み立て



—使用部品—

1. R-H-01

2. R-H-02

3. R-H-03

4. R-H-04

5. R-H-05

6. R-H-06

7. R-H-07

8. R-H-08

9. R-H-09

10.R-H-10

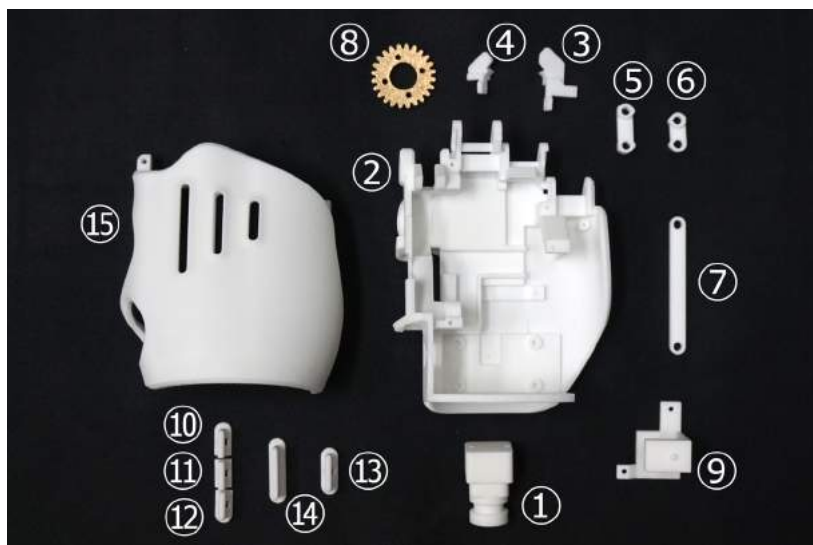
11.R-H-11

12.R-H-12

13.R-H-13

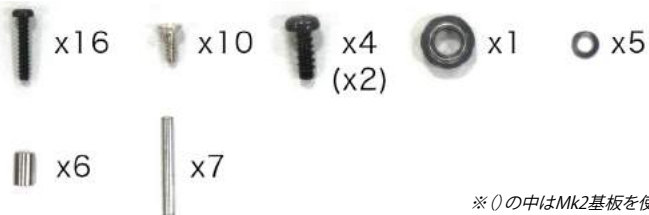
14.R-H-14

15.R-H-15



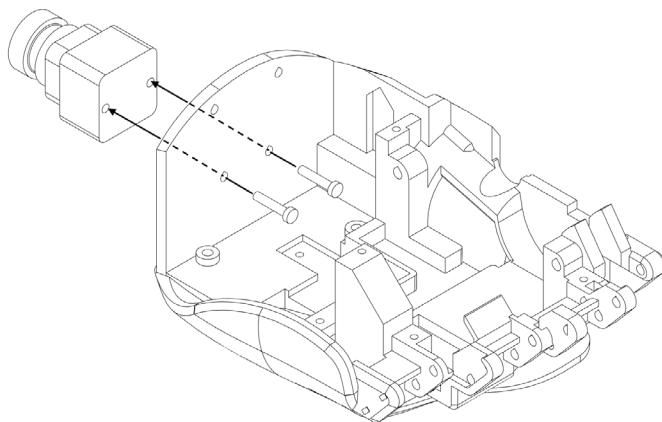
※写真のR-H-03は、強度を高めるため金属性の材料を使い3Dプリントしたものです。強度と耐久性は劣りますが、ほかの部品同様に樹脂製の材料をお使い頂くことも可能です。

—使用汎用部品—



※ ()の中はMK2基板を使用した際の数です。

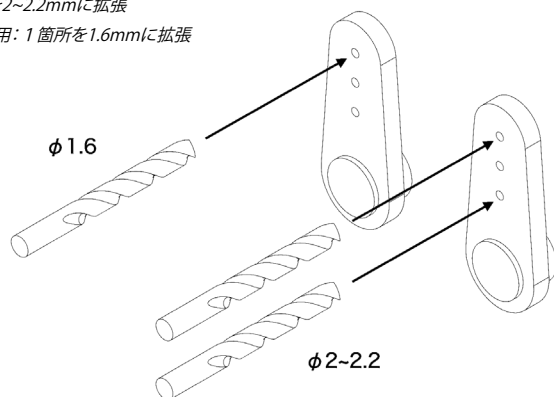
【1】2本のねじ(M2L10)で R-H-01 を R-H-02 に固定します。



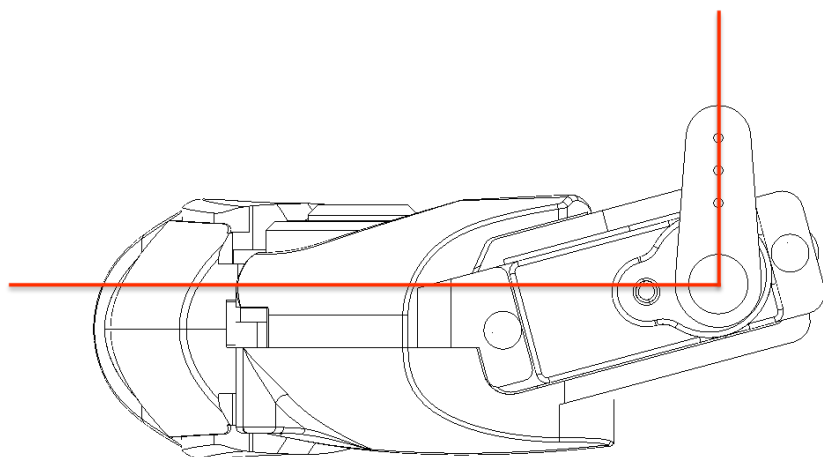
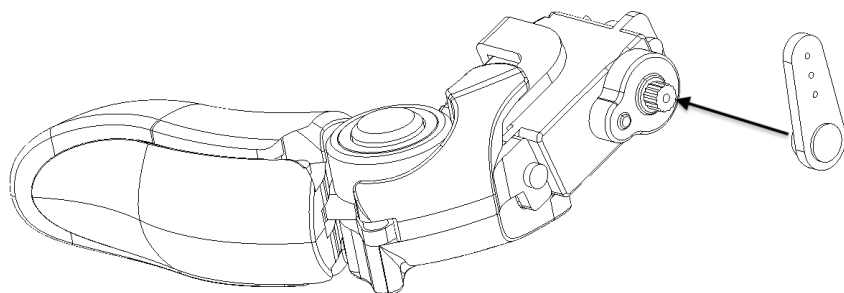
【2】サーボモーター小に付属しているサーボホーン(下図の形のもの)の穴をピンバイスを用いて拡張します。**穴の箇所と径を間違えないように注意して下さい。**

親指用: 2箇所を2~2.2mmに拡張

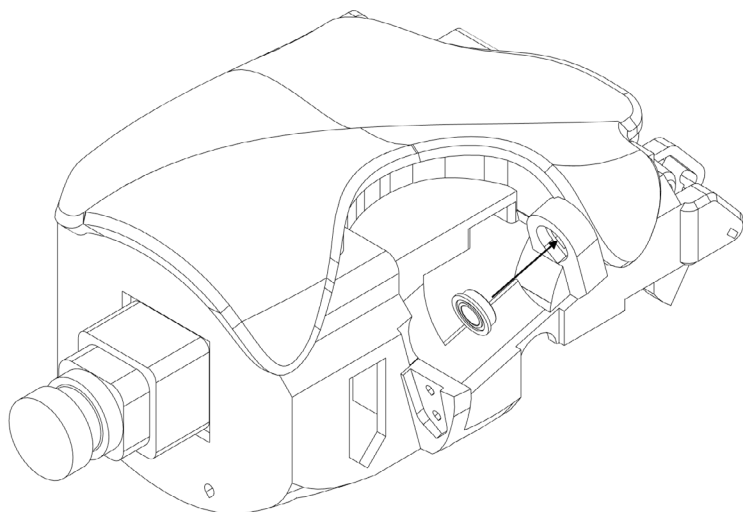
中指・薬指・小指用: 1箇所を1.6mmに拡張



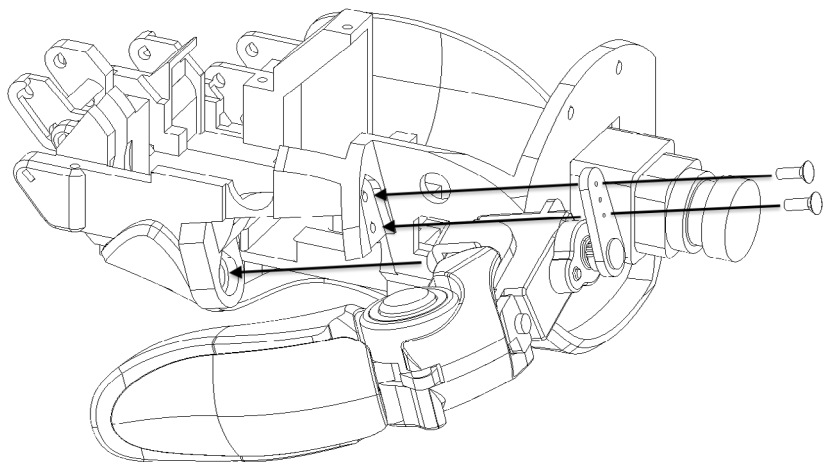
【3】サーボモーターの軸が「10 電圧の調整とサーボモーターの初期位置設定」(92ページ)で設定した状態から回転しないように注意しながら、親指ユニットとサーボホーンを直角に取り付けます。



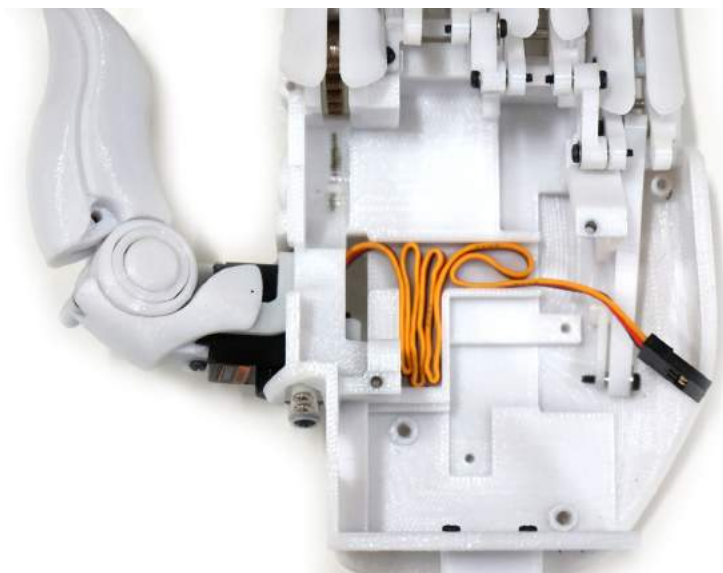
【4】ベアリングを R-H-02 に押し込みます。



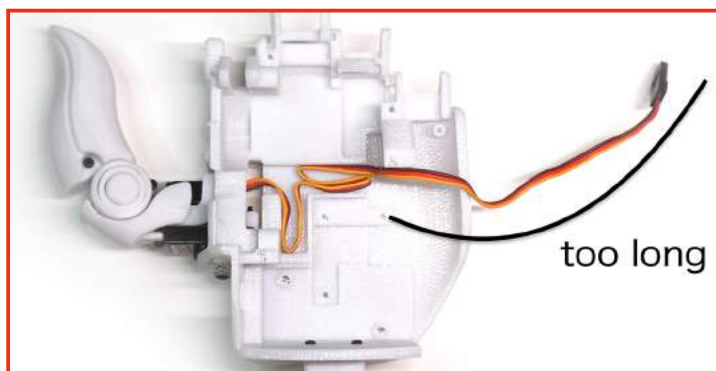
【5】親指を 2 本のねじ(M2L6)を用いて固定します。



【6】サーボモーターのケーブルは下図のよう R-H-02 に入れ込んでから折りたたみます。

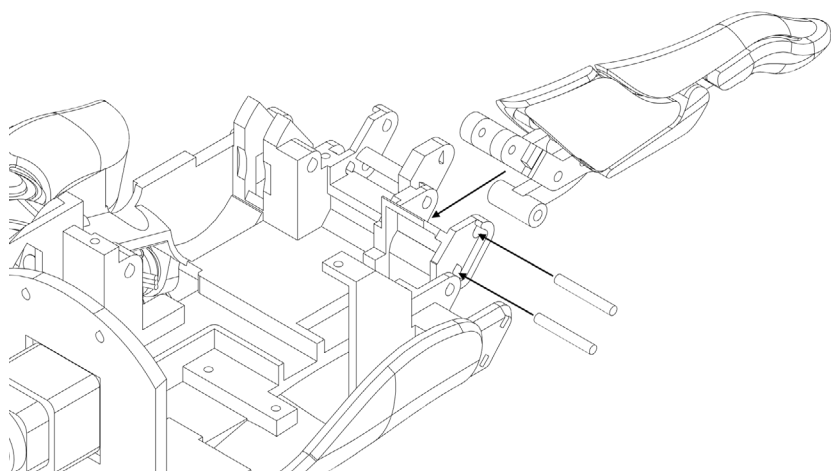


※良い例

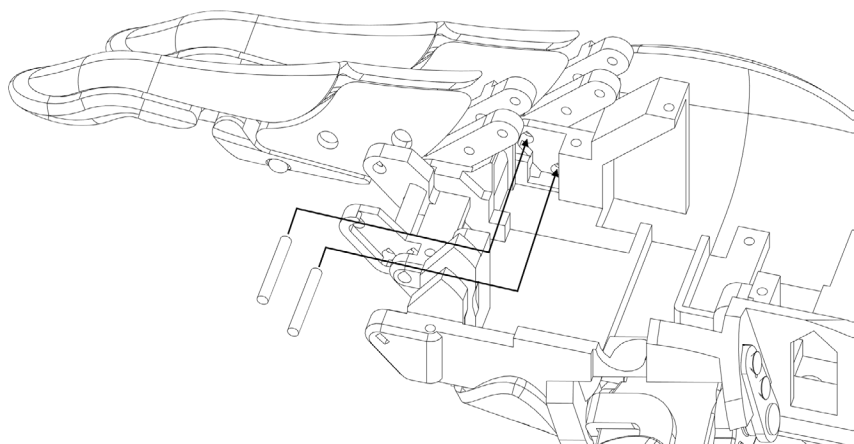


※悪い例

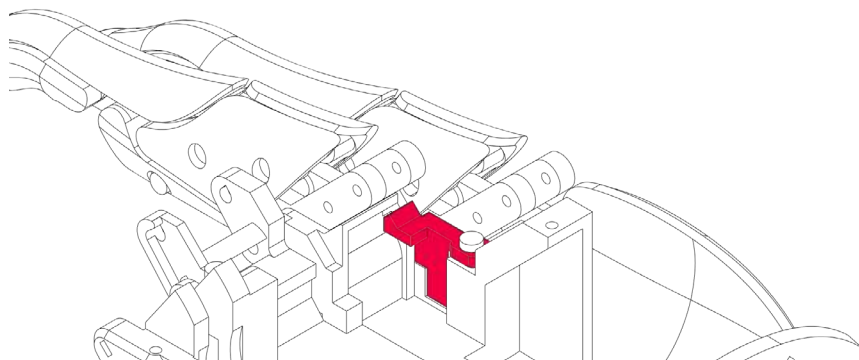
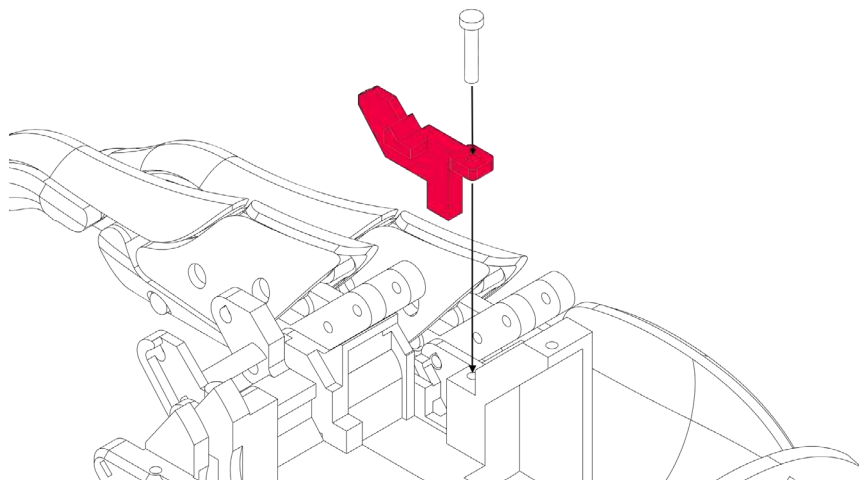
【7】次に葉指ユニットを下図のように2本のシャフト(15mm)を通して取り付けます。



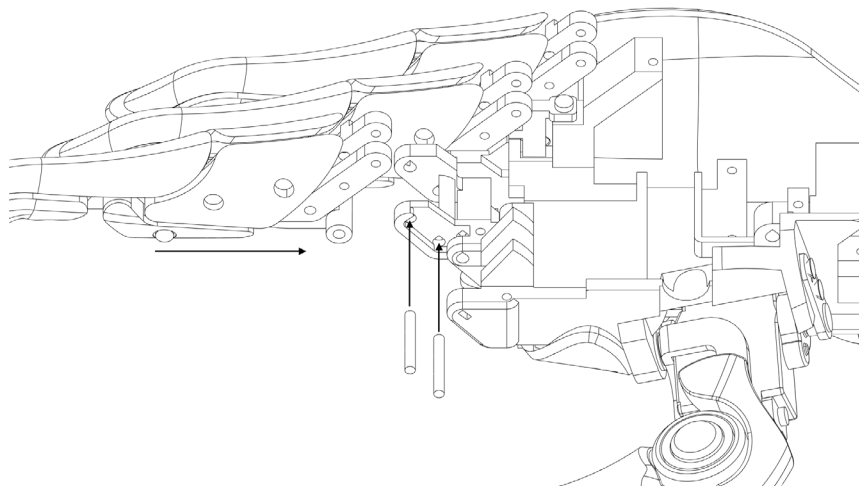
【8】同じように小指ユニットもシャフト(15mm)を2本使って取り付けます。



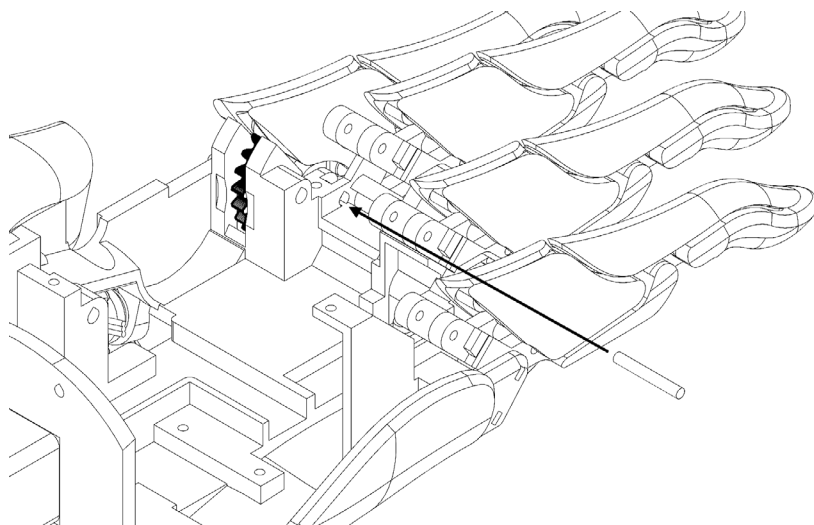
【9】ねじ(M2L10)で R-H-03 を下図のように取り付けます。



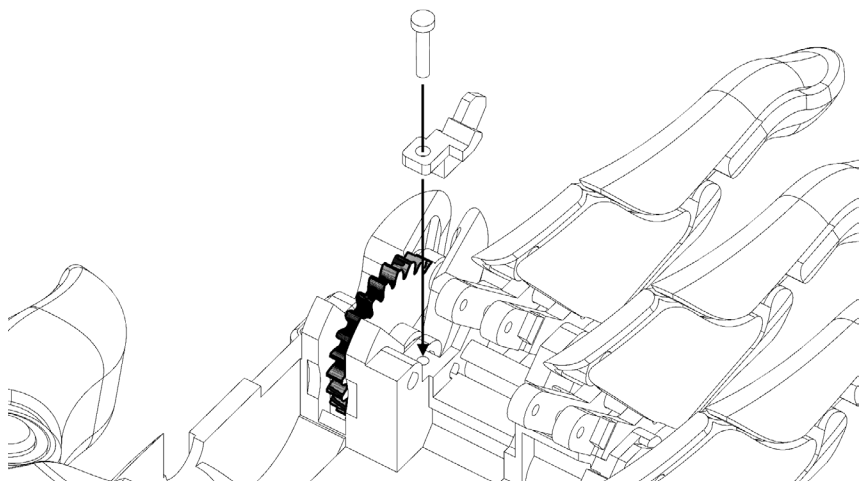
【10】中指を2本のシャフト(15mm)を用いて下図のように取り付けます。



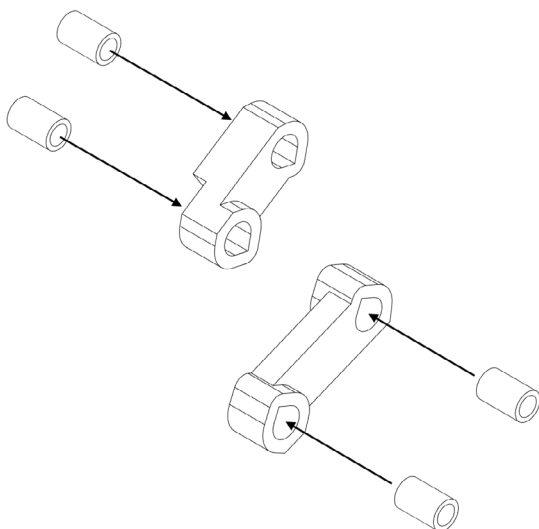
【11】人差し指ユニットをシャフト(15mm)を用いて下図のように取り付けます。この時、シャフトがギアの穴を通るようにします。



【12】R-H-04 をねじ(M2L10)を用いて下図のように取り付けます。

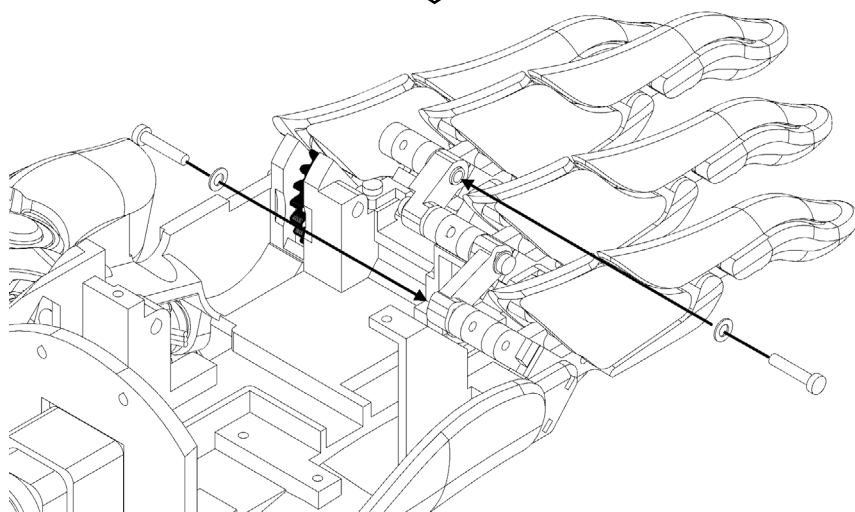
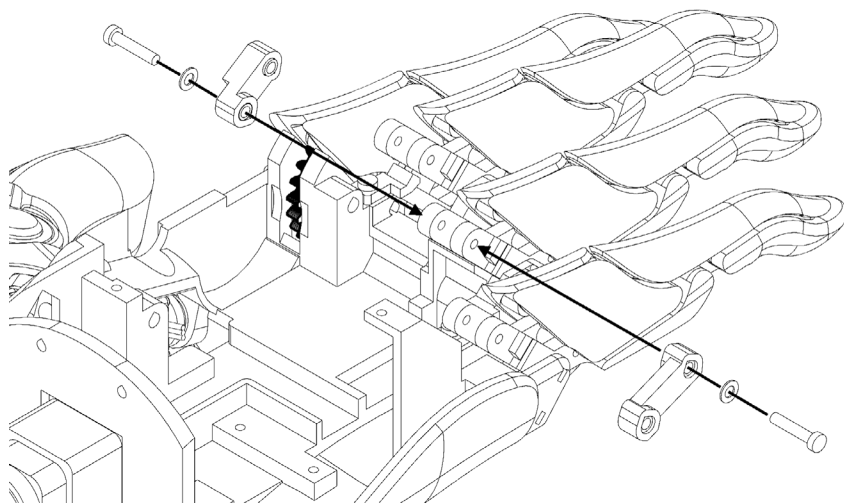


【13】R-H-05 と R-H-06 に2つずつスペーサーを入れ込みます。

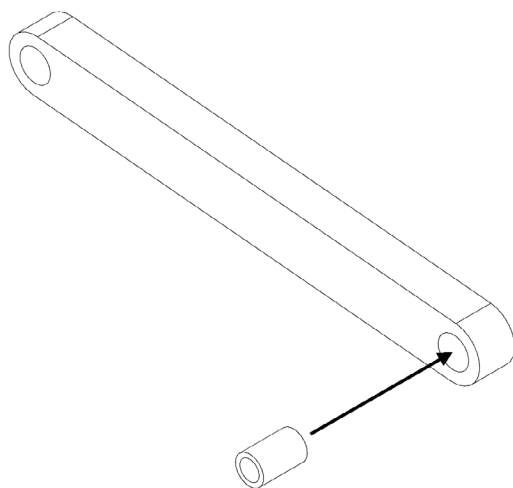


【14】R-H-05 と R-H-06 をそれぞれ 2箇所 (計 4箇所) ワッシャーを通してねじ(M2L10)で固定します。

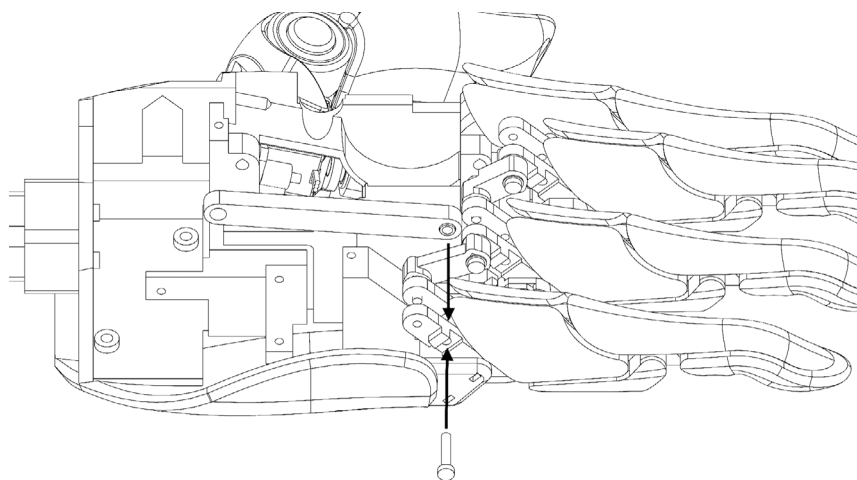
※R-H-05 (長い方の部品) が小指と薬指の間に取り付けられるように注意します。



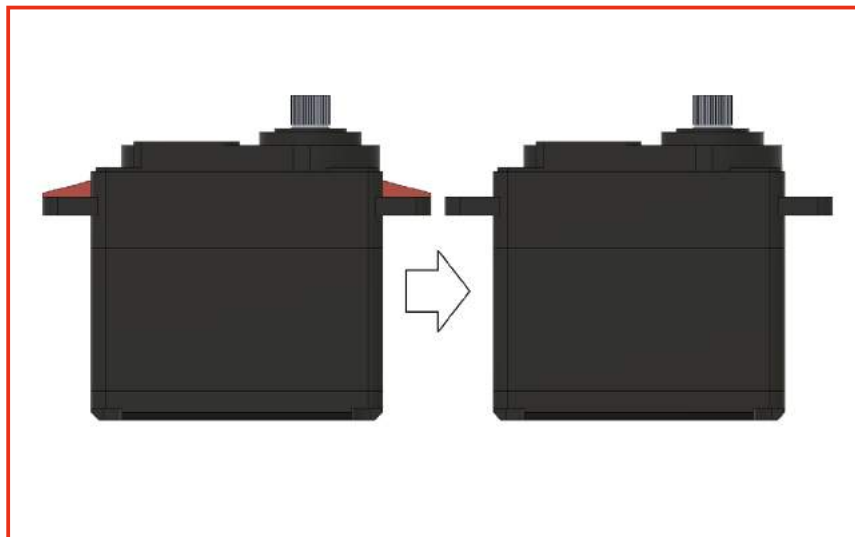
【15】スペーサーを R-H-07 のどちらかの穴に挿入します。



【16】その R-H-07 をねじM(2L10)で小指ユニットに取り付けます。

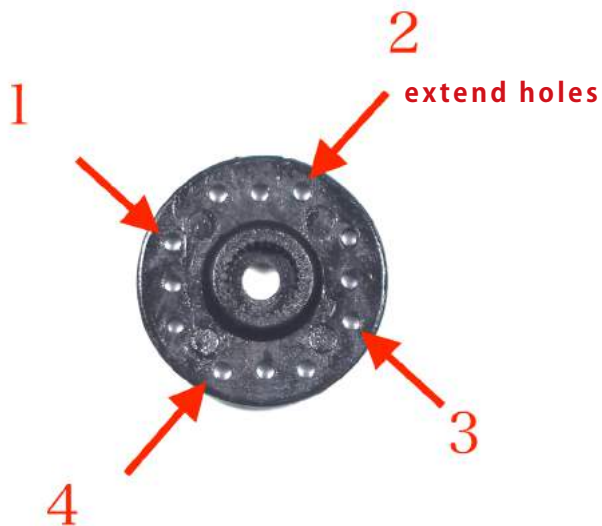


【17】サーボモーター大の下図で赤く示されている部分をニッパー等で切り取ります。

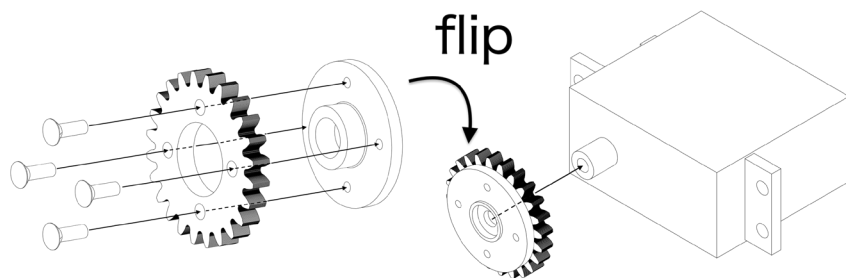


側面図

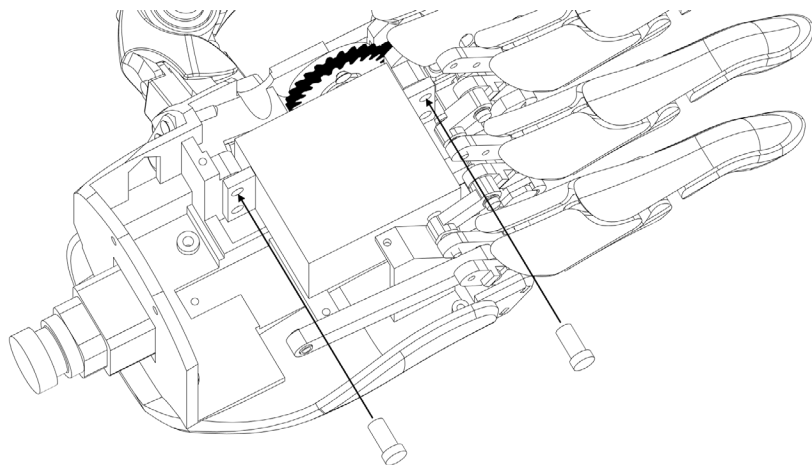
【18】サーボモーター大に同梱されているサーボホーンの穴(三連の穴の右端)を4箇所ピンバイスを用いて1.6mmに拡張します。



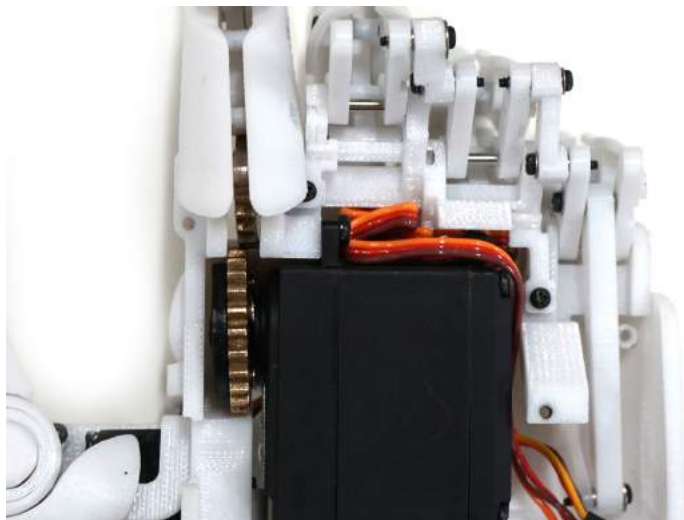
【19】R-H-08 を4本のねじ(M2L6)を使ってサーボホーンに取り付けます。



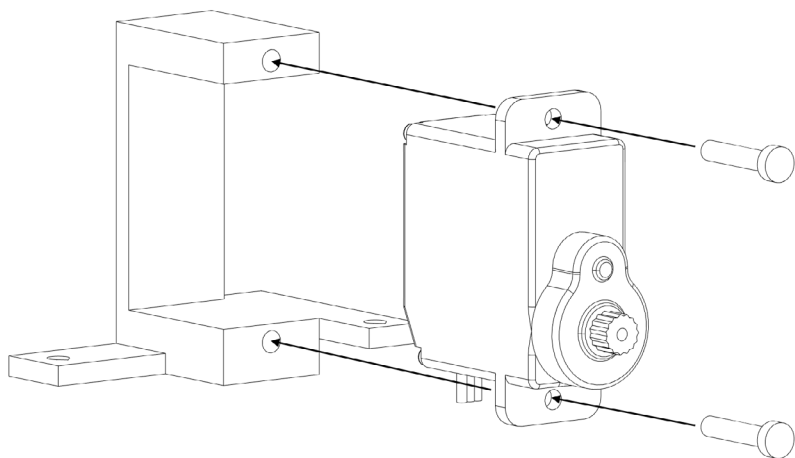
【20】サーボモーターの軸を回転させないように注意しつつ、人差し指が伸びた状態で R-H-02 にサーボモーター大を 2 本のねじ(M3L8)で取り付けます。



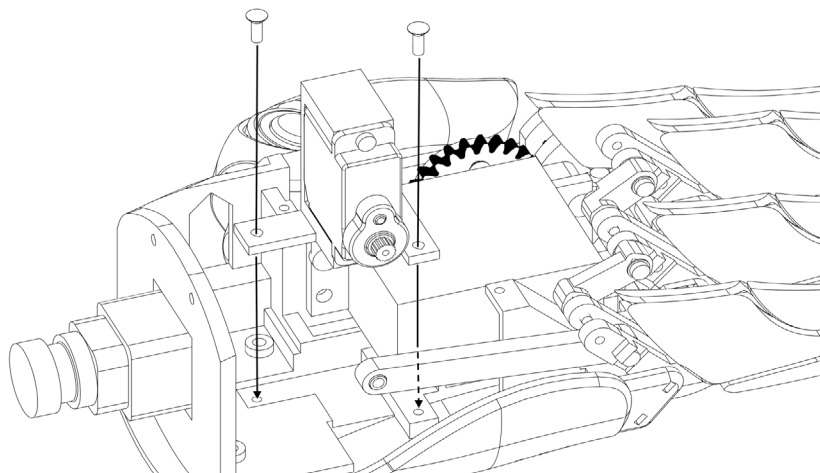
【21】下図のようにケーブルをまとめます。

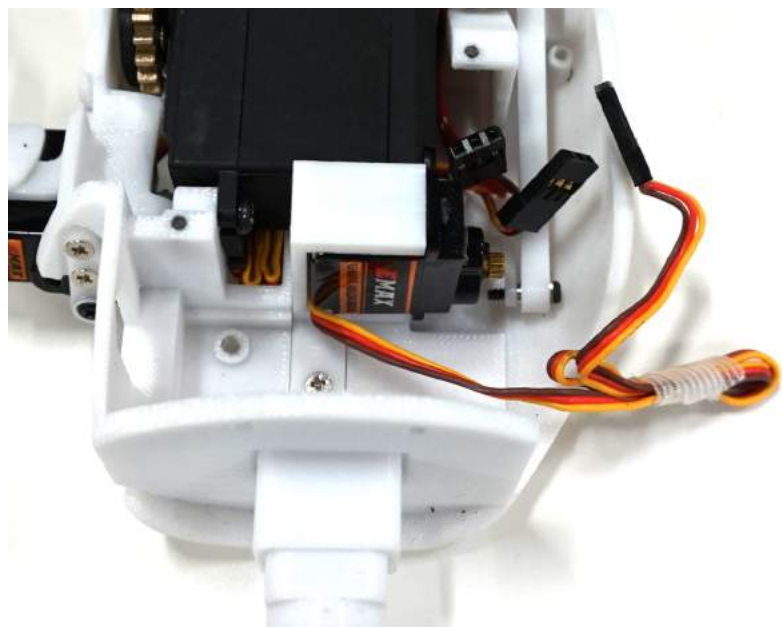


【22】サーボモーター小を R-H-09 に2本のねじ(M2L10)で固定します。この時、ケーブルは左側に出すよう注意してください。

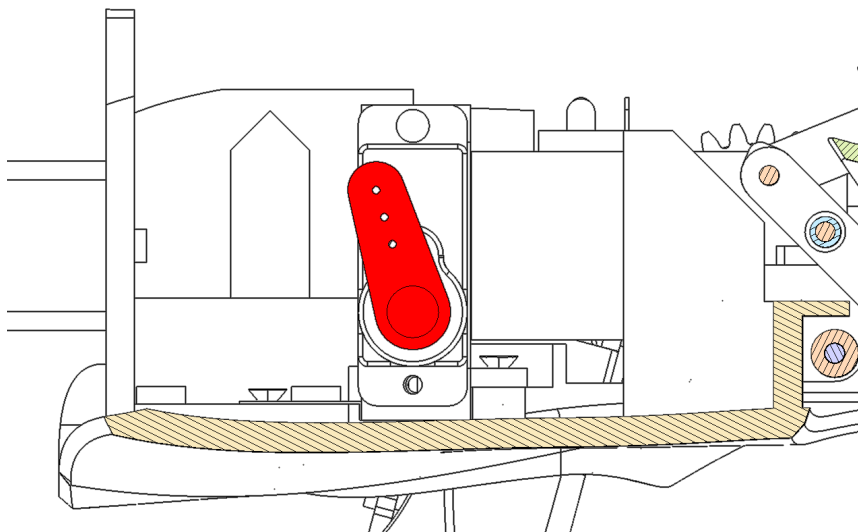
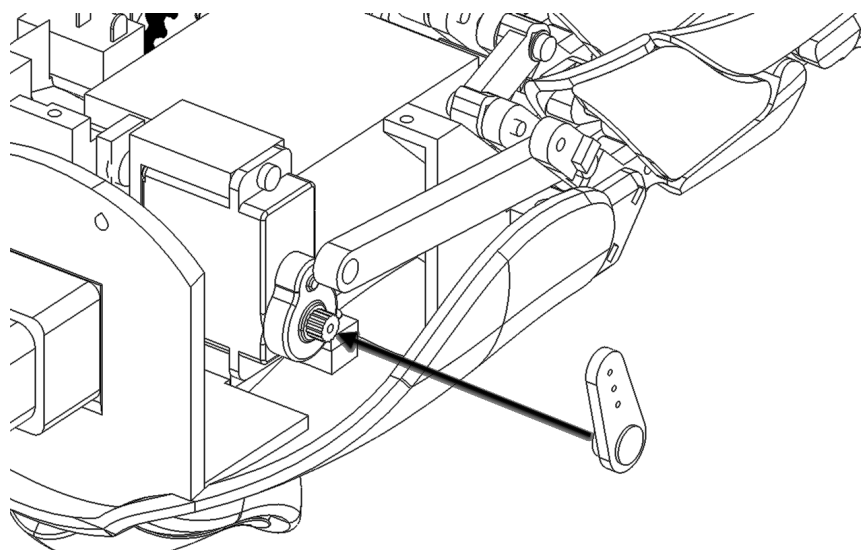


【23】これを2本のねじ(M2L6)で固定し、2枚目の写真のようにケーブルを取り回します。ここで誤ってねじ(M2L10)を使うとねじが掌から突き出してしまうので十分注意してください。

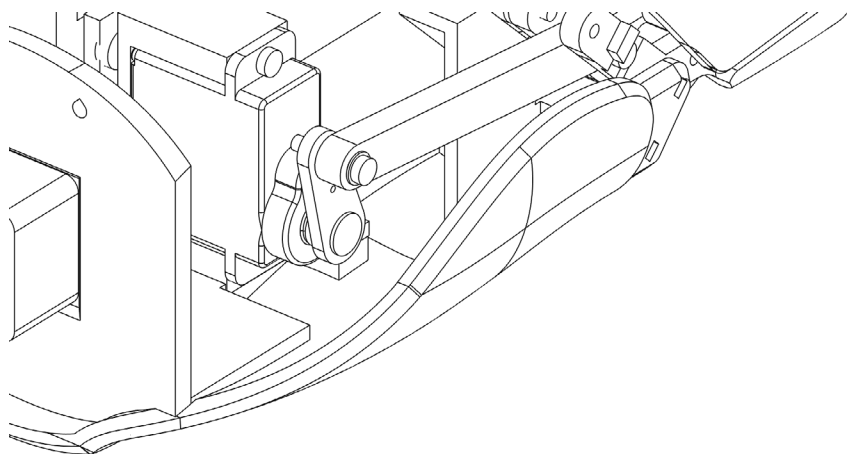
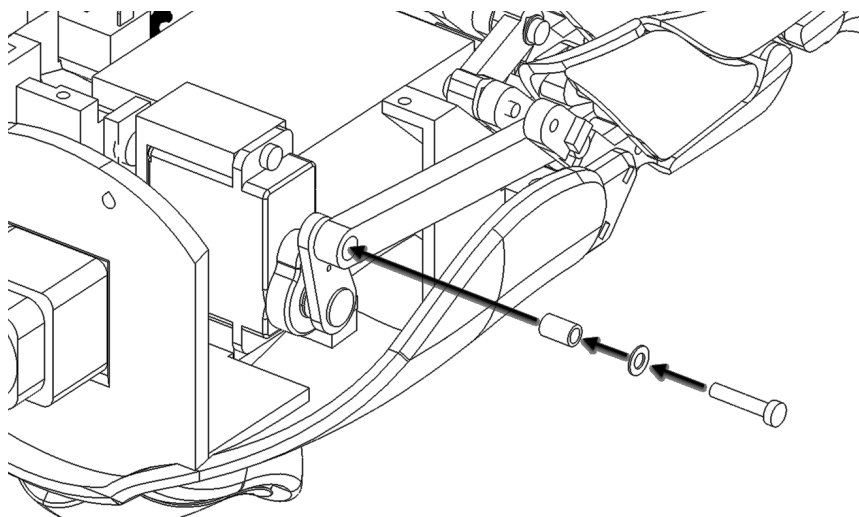




【24】サーボモーターの軸が「10 電圧の調整とサーボモーターの初期位置設定」(92ページ)で設定した状態から回転しないように注意しながら、サーボモーターにサーボホーンを取り付けます。



【25】サーボホーンに R-H-07 をワッシャーとスペーサーを通しねじ(M2L10)で取り付けます。



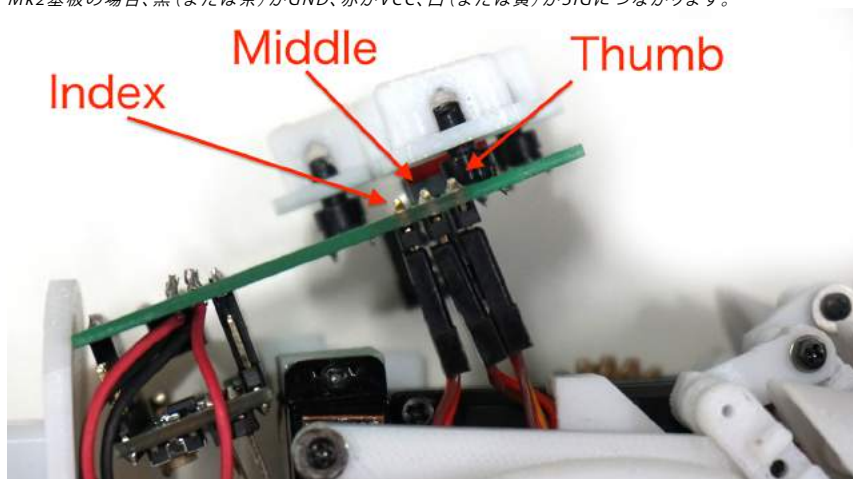
【26】サーボモーターのケーブルは写真のように通します。



【27】基板上のピンヘッダにサーボモーターのコネクタを指します。黒(または茶)のケーブルが外側になるよう注意してください。Indexと書かれたピンヘッダにはサーボモーター大、MiddleおよびThumbと書かれたピンヘッダにはサーボモーター小のコネクタを差します。

Mk1基板の場合、黒(または茶)がGND、赤がV+、白(または黄)がPWMにつながります。

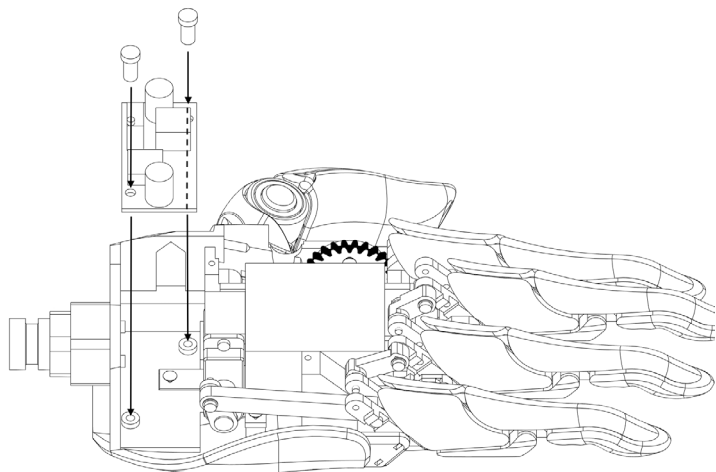
Mk2基板の場合、黒(または茶)がGND、赤がVCC、白(または黄)がSIGにつながります。



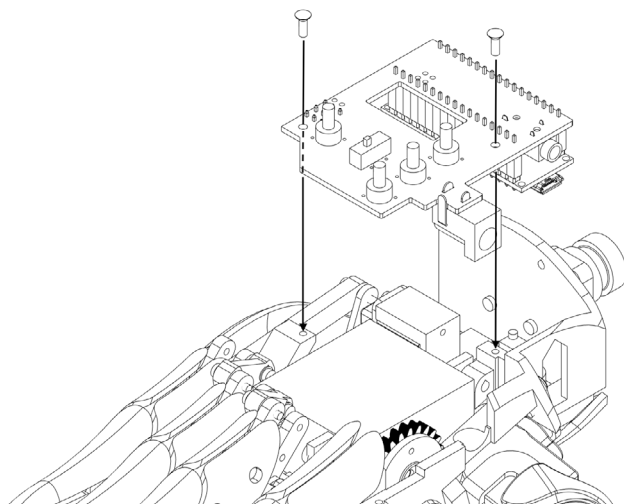
≫次に基板の取り付けです。

Mk1の場合は【28】から、Mk2基板の場合は【33】以降を参照してください。

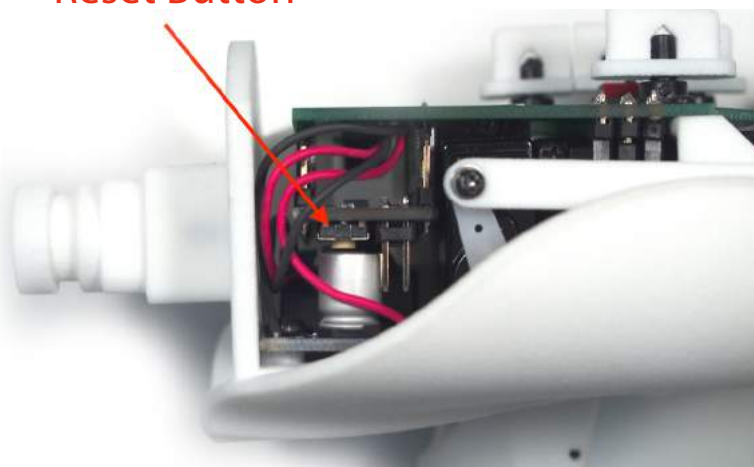
【28】DCDCコンバーターを2本のねじ(M3L8)で固定します。



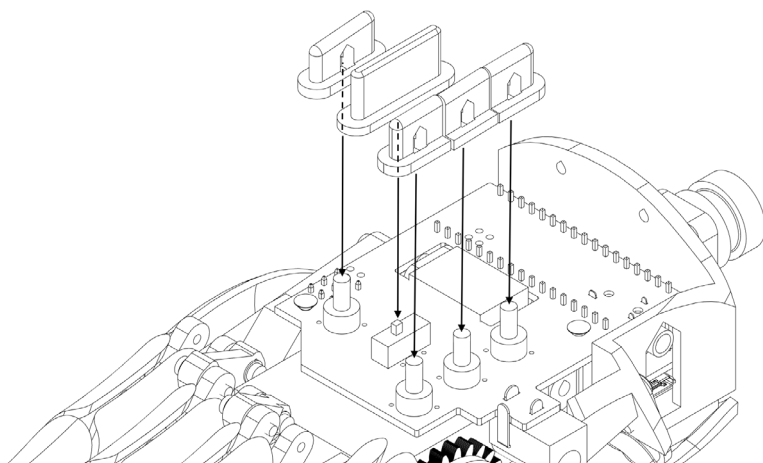
【29】基板を2本のねじ(M2L6)を用いて固定します。この時、Arduino Microが奥まで挿さって
いなかったり、基板固定用のねじを強く締めすぎると、Arduino Microのリセットボタンが
HACKberryが正しく動作しないので注意してください。



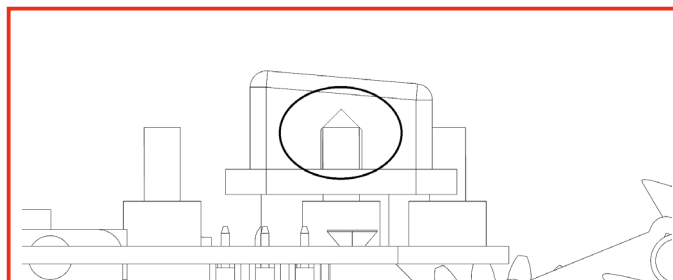
Reset Button



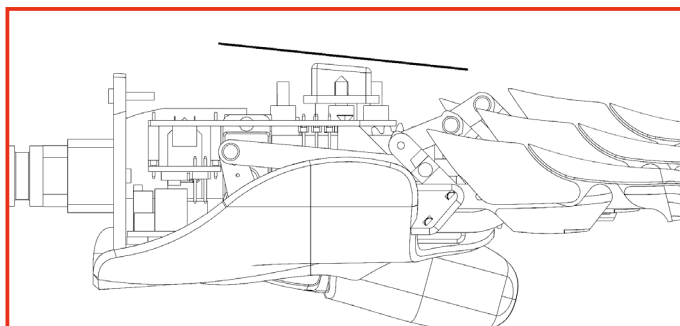
【30】R-H-10, R-H-11,R-H-012,R-H-13,R-H-14を下図のように取り付けます。



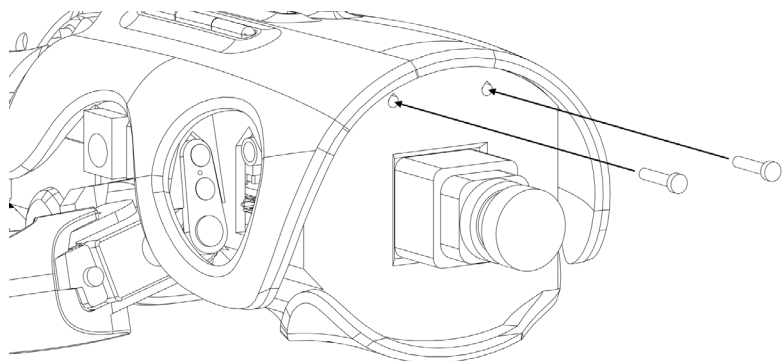
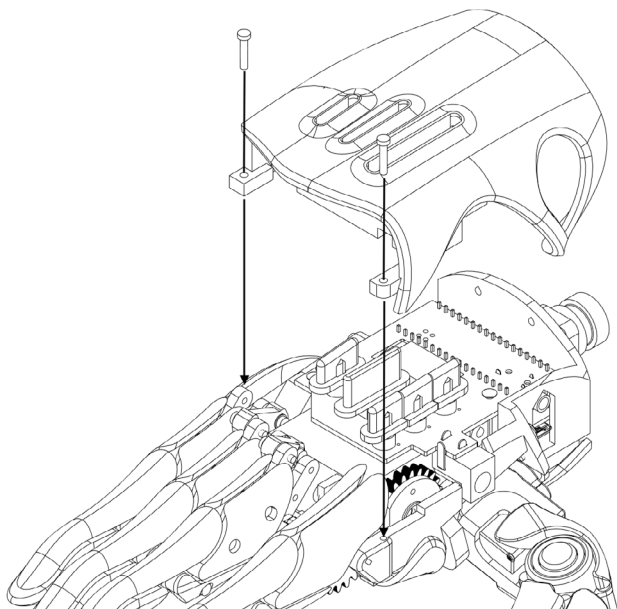
※全て奥までしっかりと挿入してください。



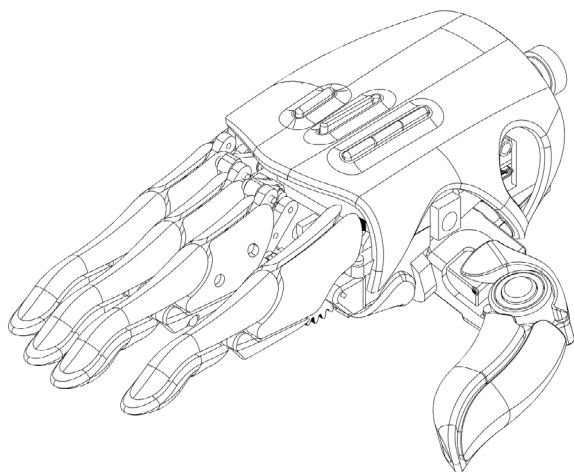
※R-H-13 は背の低い側が指先を向くようにします。



【31】R-H-15 を4本のねじ(M2L10)を用いて固定します。この時指を曲げていると取り付けやすいです。

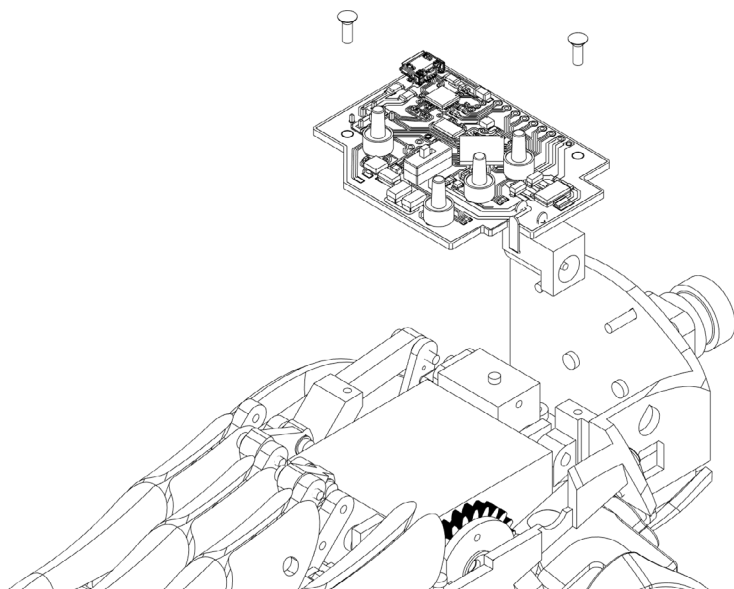


【32】これで掌の組み立ては完了です。次の章に進んでください。

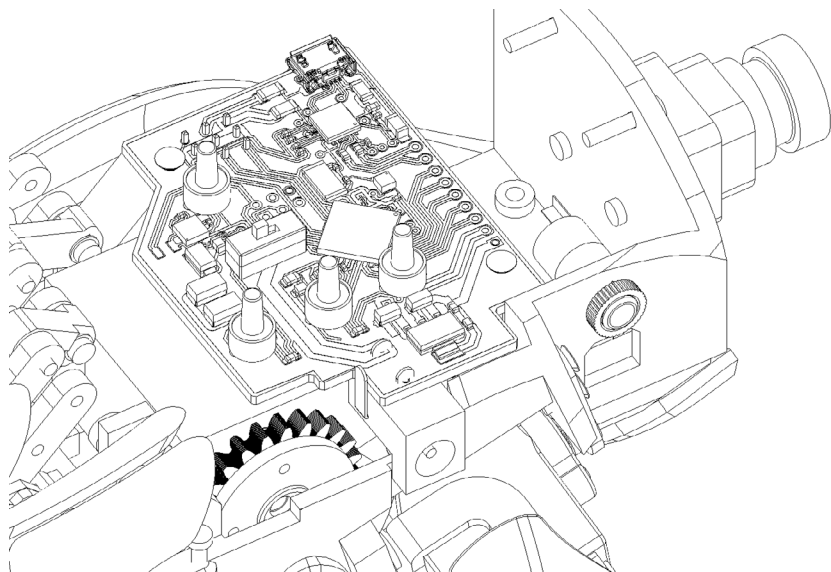
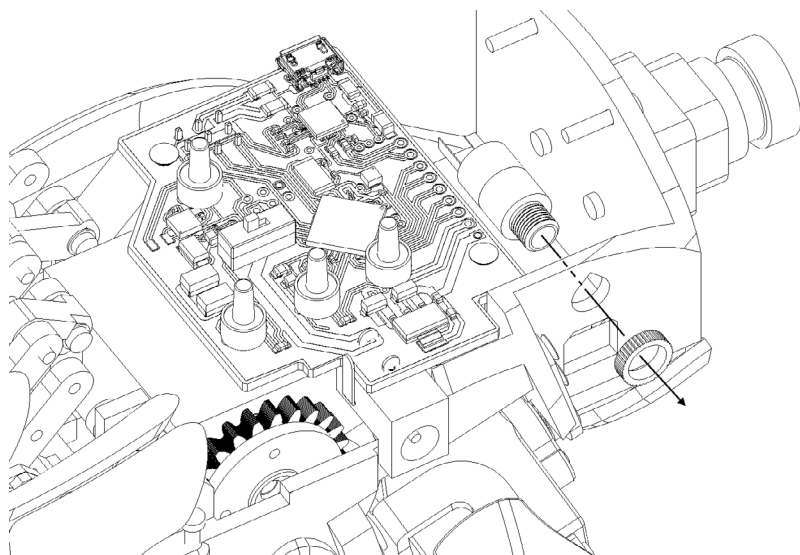


≫ Mk2基板の場合はここからです。

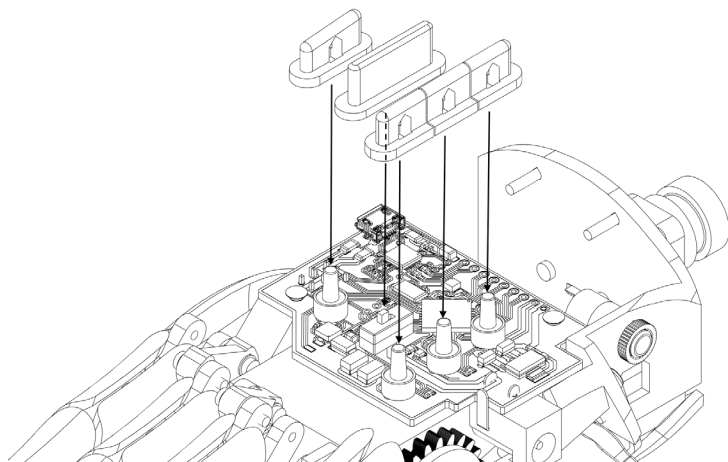
【33】HACKberry hand board Mk2を2本のねじ(M2L6)を用いて固定します。



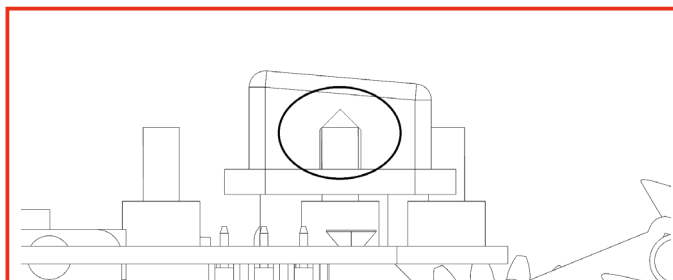
【34】ステレオジャックのナットを一度外し、その後R-H-01を挟むように再度取り付けます。



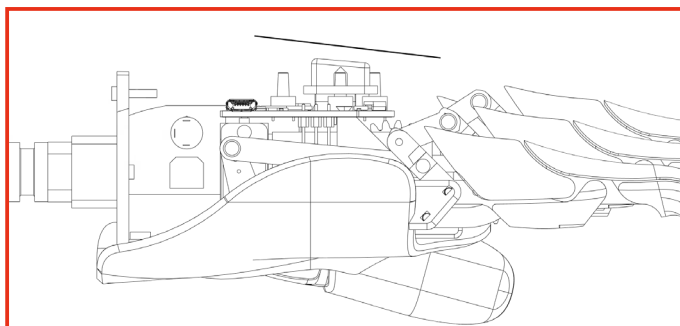
【35】R-H-10, R-H-11, R-H-012, R-H-13, R-H-14を下図のように取り付けます。



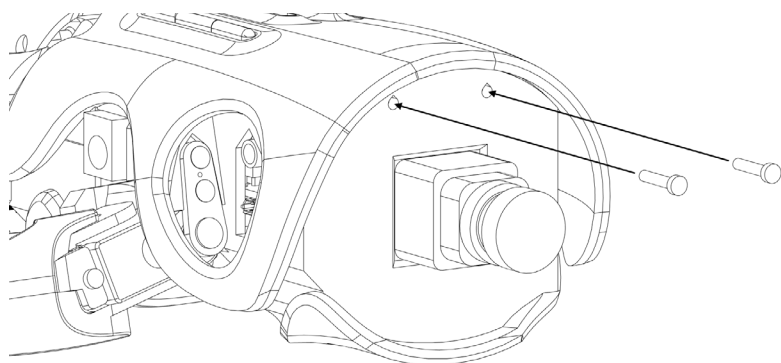
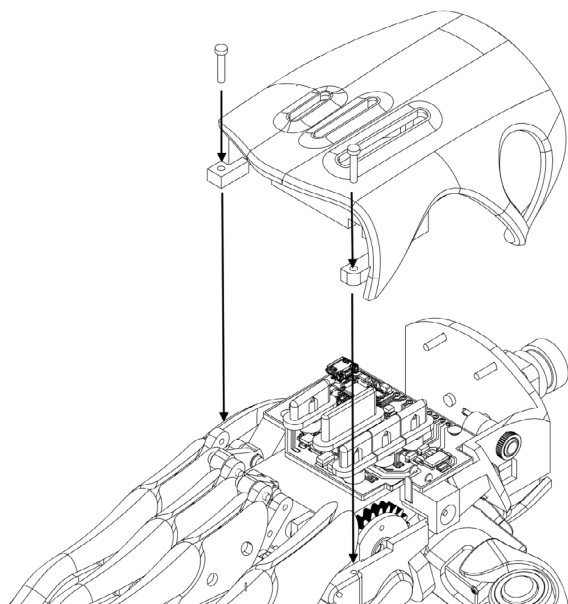
※全て奥までしっかりと挿入してください。



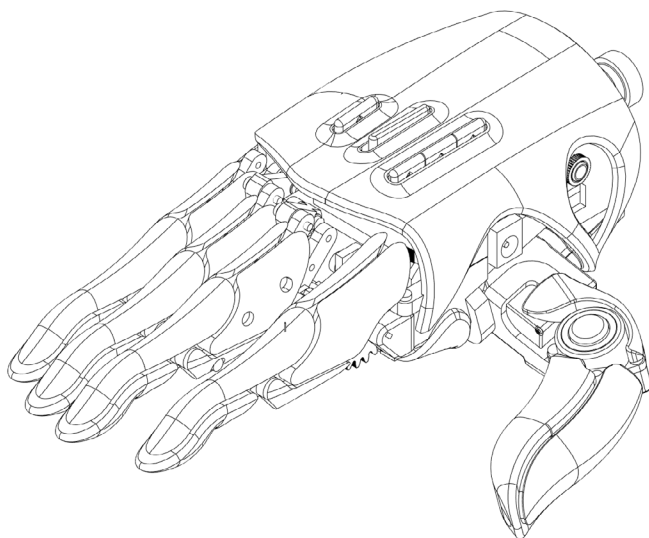
※R-H-13は背の低い側が指先を向くようにします。



【36】R-H-15を4本のねじ(M2L10)を用いて固定します。この時指を曲げていると取り付けやすいです。



【47】これで掌の組み立ては完了です。



Assembling

11 手首の組み立て

※本項では手首ユニットの組み立てを行います。ダミーソケット無しの場合とダミーソケット有りの場合で手首ユニットに接続される部品が異なるので注意してください。

ダミーソケット無しの場合



ダミーソケット有りの場合



—使用部品—

1. R-W-01

4. R-W-04

7. R-W-07

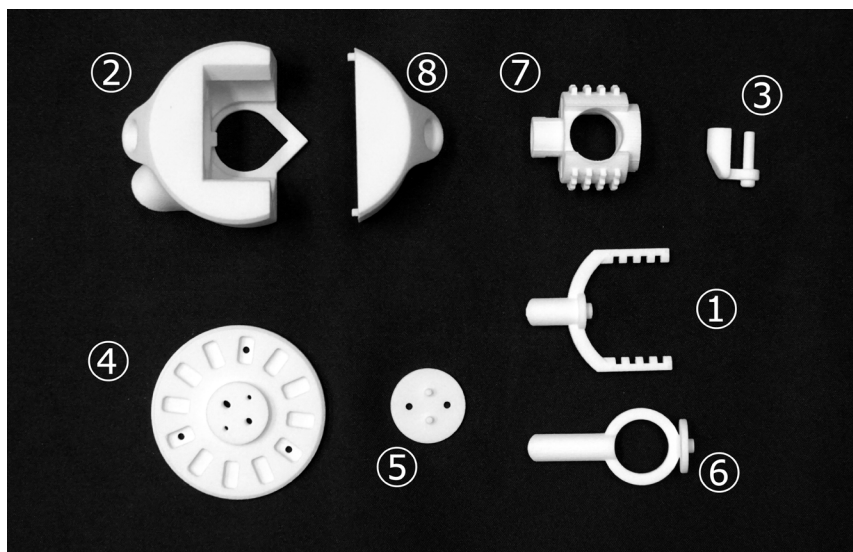
2. R-W-02

5. R-W-05

8. R-W-08

3. R-W-03

6. R-W-06



—使用汎用部品—

 x3
(x0)

 x2

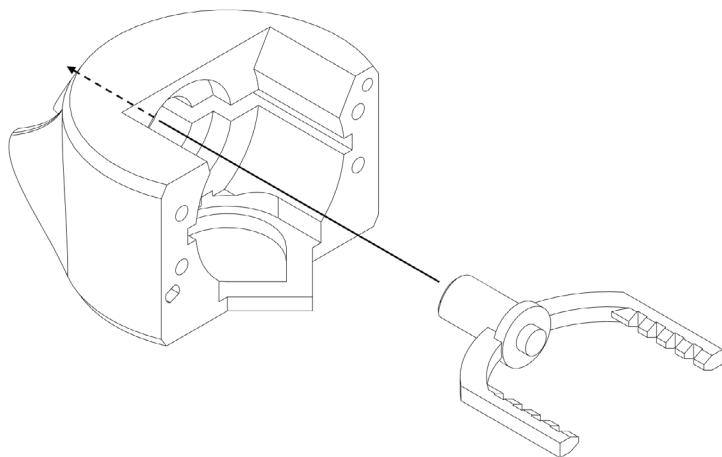
 x2

 x4

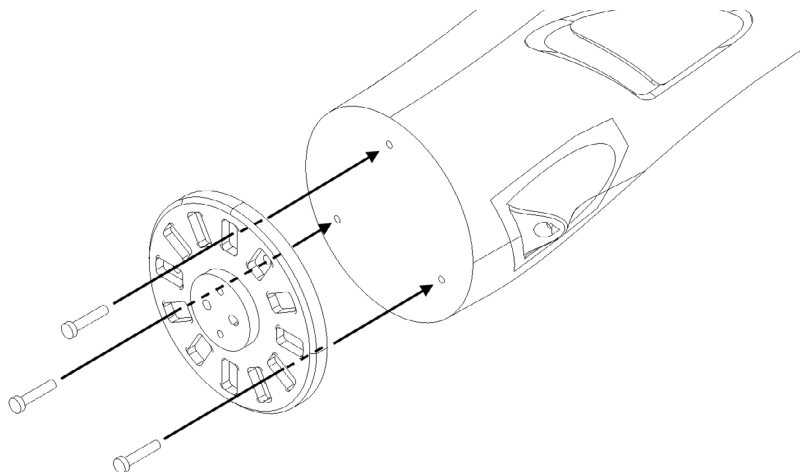
 x2

※ 0の中はダミーソケット無しの場合の数量です。

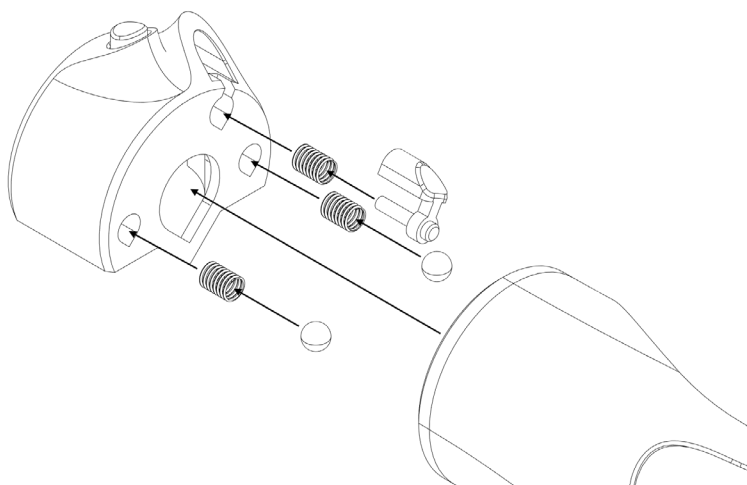
【1】R-W-01 を R-W-02 に差し込みます。R-W-01 のボタン面は斜めになっていますが、長い方が下になります。



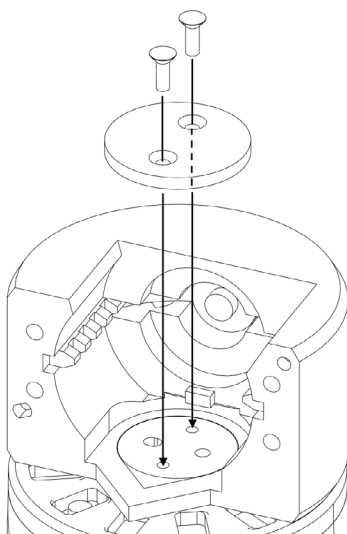
【2】(ダミーソケット無しの場合、この工程は不要です) 3本のねじ(M2L10)を用いてR-W-04にダミーソケットを取り付けます。



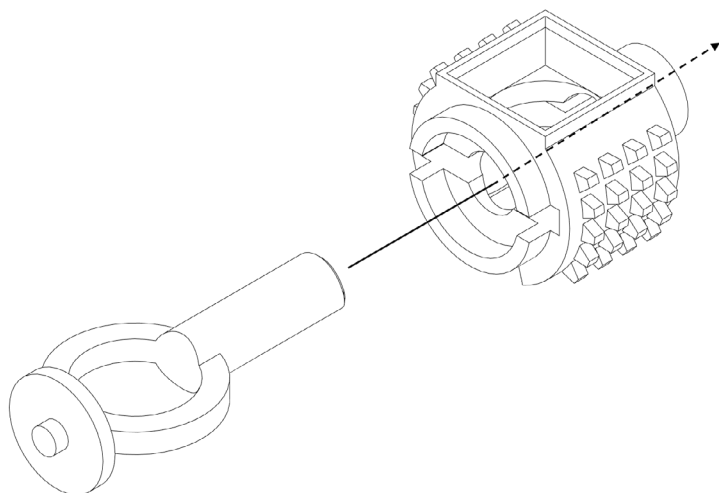
【3】R-W-02 と R-W-04 で 3 つのコイルばねと 2 つの鋼球、R-W-03 を挟み込みます。



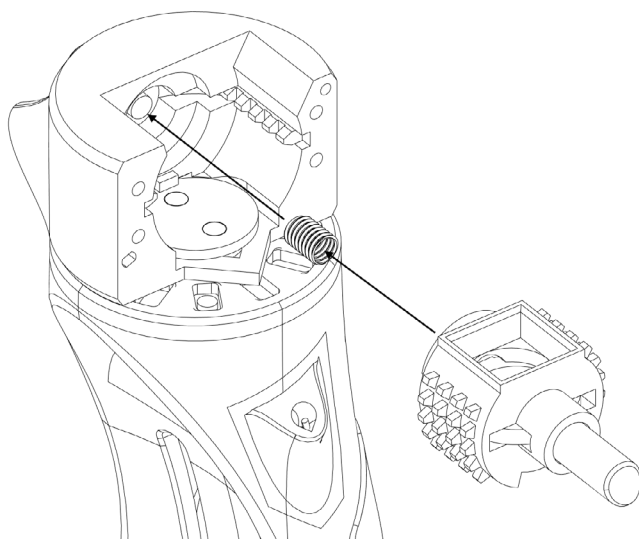
【4】R-W-04とR-W-05 を 2 本のねじ(M2L6)で固定します。



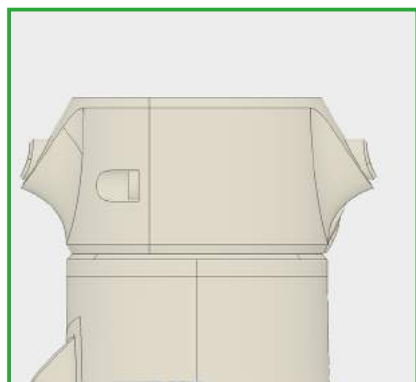
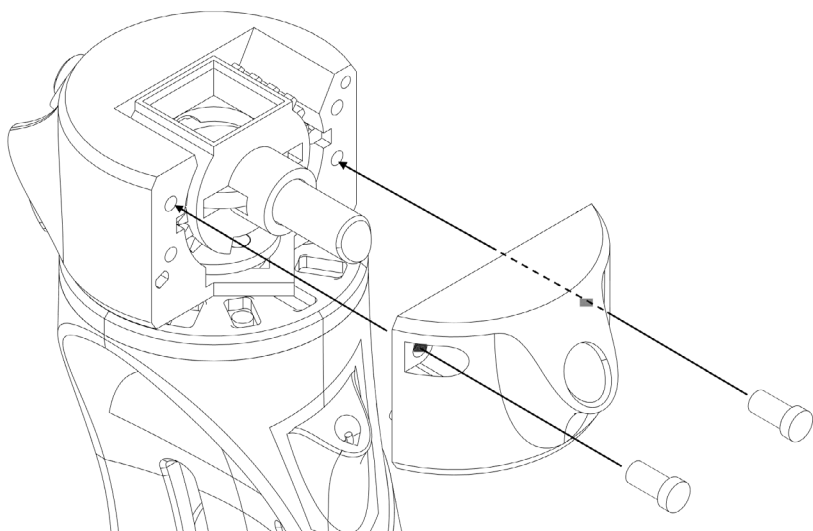
【5】R-W-06 を R-W-07 に差し込みます。R-W-06のボタン面は斜めになっていますが、長い方が下になります。



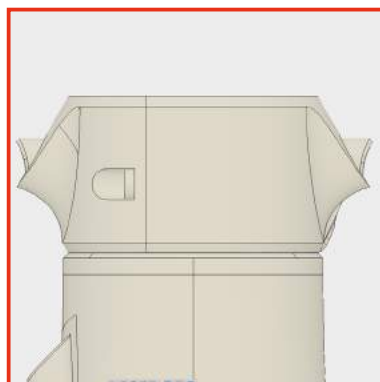
【6】コイルばねと先ほど組んだ部品を図のように挿入します。



【7】R-W-08 を取り付け、2本のねじ(M3L8)で固定します。

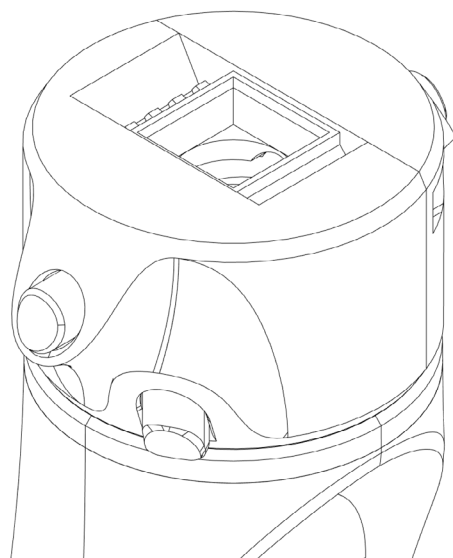


※良い例



※悪い例

【8】 これで手首ユニットの組み立ては終了です。



Assembling

13 センサの組み立て



—使用部品—

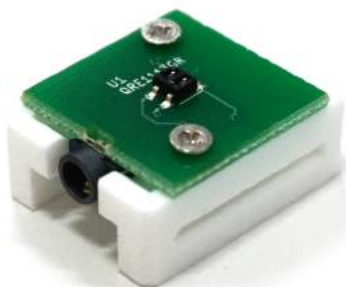
- | | | |
|--------------|-------------|-----------------|
| 1. C-SE-01 | 2. C-SE-02 | 3. Sensor board |
| 4. ベルクロストラップ | 5. ステレオケーブル | 6. センサクッション |



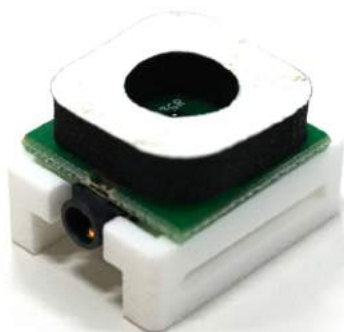
—使用汎用部品—



【1】Sensor boardをC-SE-01に写真の向きで組み付け、2本のねじ(M2L6)で固定します。



【2】センサクションの剥離紙を剥がして写真のようにSensor boardに貼り付けます。



【3】反対側の剥離紙を剥がし、写真のようにC-SE-02をセンサクションに貼り付けます。



【2】写真のようにC-SE-01にベルクロストラップを通します。これでセンサの組み立ては終わりです。



Assembling

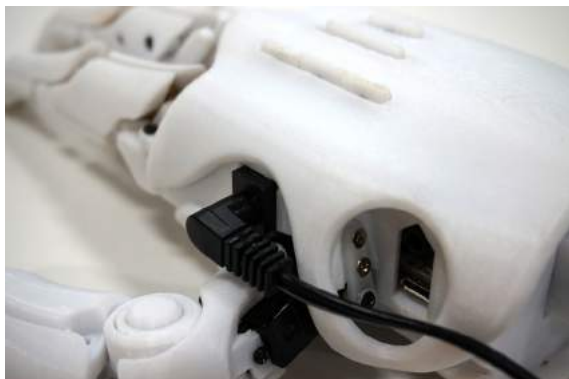
14 初期位置設定の方法

※本項では、モーターの初期位置設定を行います。モーターの個体差や取付時のずれにより、指の角度が若干ずれている場合があります、それをプログラム内の値を変更して調整します。

右手の場合は本ページから、左手の場合は139ページから進めてください。

右手の場合

【1】バッテリーの入ったバッテリーケースもしくはダミーソケットのDCプラグを掌ユニットのDCジャックに挿し込みます。



【2】手の甲の真ん中のボタンを押し上げ、HACKberryの電源を入れます。電源投入直後は図のように親指が手の側面に開き、他の指は開いた状態になります。しかし、モーターの個体差や取付時のずれにより、指が伸びていなかったり、逆に過度に伸ばそうとして「ジジジ」という音が鳴り続けてしまう場合があります。その場合はすぐに電源を切ります。

※Mk2基板はMk1基板と比較して電源投入からサーボモータが動き出すまでの時間が少し長いです。



本項では一例として、下図のように人差し指が曲がりすぎ、一方で、親指が開きすぎてしまった場合を考えます。



【3】「5 プログラムのアップロード」(52ページ)で使用したプログラムを開きます。指の角度の調節は図に示す6つの数字を変更して行います。

```
Hackberry3.1
1 |
2 * Arduino micro code for HACKBerry.
3 * Originally created by ex111 Inc.
4 * edited by Genta Kondo on 2017/6/11
5 */
6 #include <Servo.h>
7
8 //Settings
9 const boolean isRight = 0;//right:1, left:0
10
11 //For right hand, find optimal values of ThumbMin, IndexMax and OtherMax first.
12 //For left hand, find optimal values of ThumbMax, IndexMin and OtherMin first.
13 //Then, calculate the remaining values by following rules.
14 //Difference of ThumbMin and ThumbMax is 86
15 //Difference of IndexMin and IndexMax is 117
16 //Difference of OtherMin and OtherMax is 55
17
18 const int outThumbMax = 150;//right:open, left:close
19 const int outIndexMax = 150;//right:open, left:close
20 const int outOtherMax = 150;//right:open, left:close
21
22 const int outThumbMin = 30;//right:close, left:open
23 const int outIndexMin = 30;//right:close, left:open
24 const int outOtherMin = 30;//right:close, left:open
25
26 const int speedMax = 6;
27 const int speedMin = 0;
28 const int speedReverse = -3;
29 const int thSpeedReverse = 15;//0-100
```

【4】Maxと付いた値が指が開いた角度、Minと付いた値が指が閉じた角度を示しています。今回、開いていない状態で人差し指が曲がっていたので、より開くためにMaxの値を少しずつ大きくしていきます。

今回の場合、

```
const int outIndexMax = 150;
```

だったものを

```
const int outIndexMax = 161;
```

にしてプログラムをアップロードした時、指が正しい角度になりました。

```
17  
18 const int outThumbMax = 150;//right:open, left:close  
19 const int outIndexMax = 161;//right:open, left:close  
20 const int outOtherMax = 150;//right:open, left:close  
21  
22 const int outThumbMin = 30;//right:close, left:open  
23 const int outIndexMin = 30;//right:close, left:open  
24 const int outOtherMin = 30;//right:close, left:open  
25
```



【5】次に親指の調整です。親指は開きすぎている状態なので、Maxの値を少しずつ小さくしていきます。

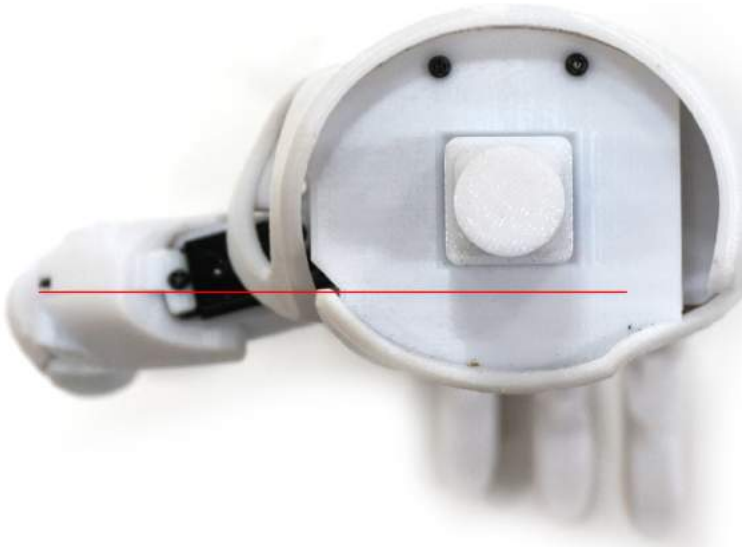
```
const int outThumbMax = 150;
```

だったものを

```
const int outThumbMax = 136;
```

にしてプログラムをアップロードした時、指が正しい角度になりました。

```
17  
18 const int outThumbMax = 136; //right:open, left:close  
19 const int outIndexMax = 161; //right:open, left:close  
20 const int outOtherMax = 150; //right:open, left:close  
21  
22 const int outThumbMin = 30; //right:close, left:open  
23 const int outIndexMin = 30; //right:close, left:open  
24 const int outOtherMin = 30; //right:close, left:open  
25
```



【6】次に指が閉じた状態の値を設定します。閉じた状態と開いた状態の角度差は、それぞれのモーターにおいて、次の値を参考にしてください。

```
-thumb: 86  
-index: 117  
-other: 55
```

例えば、先程親指の開いた状態の角度を136と設定したので、136-86 と記入します。
他の値も同様に設定しプログラムをアップロードします。

```
21  
22 const int outThumbMin = 136-86;//right:close, left:open  
23 const int outIndexMin = 161-117;//right:close, left:open  
24 const int outOtherMin = 150-55;//right:close, left:open  
25
```

左手の場合

【1】バッテリーの入ったバッテリーケースもしくはソケットのDCプラグを掌ユニットのDCジャックに挿し込みます。



【2】手の甲の真ん中のボタンを押し上げ、HACKberryの電源を入れます。電源投入直後は図のように親指が手の側面に開き、他の指は開いた状態になります。しかし、モーターの個体差や取付時のずれにより、指が伸びていなかったり、逆に過度に伸ばそうとして「ジジジ」という音が鳴り続けてしまう場合があります。その場合はすぐに電源を切ります。



本項では一例として、下図のように人差し指が曲がりすぎ、一方で、親指が開きすぎてしまった場合を考えます。



【3】「5 プログラムのアップロード」(52ページ)で使用したプログラムを開きます。指の角度の調節は図に示す6つの数字を変更して行います。

```
Hackberry3.1
1 //
2 * Arduino micro code for HACKBerry.
3 * Originally created by ex111 Inc.
4 * edited by Genta Kondo on 2017/6/11
5 */
6 #include <Servo.h>
7
8 //Settings
9 const boolean isRight = 0; //right:1, left:0
10
11 //For right hand, find optimal values of ThumbMin, IndexMax and OtherMax first.
12 //For left hand, find optimal values of ThumbMax, IndexMin and OtherMin first.
13 //Then, calculate the remaining values by following rules.
14 //Difference of ThumbMin and ThumbMax is 86
15 //Difference of IndexMin and IndexMax is 117
16 //Difference of OtherMin and OtherMax is 55
17
18 const int outThumbMax = 150; //right:open, left:close
19 const int outIndexMax = 150; //right:open, left:close
20 const int outOtherMax = 150; //right:open, left:close
21
22 const int outThumbMin = 30; //right:close, left:open
23 const int outIndexMin = 30; //right:close, left:open
24 const int outOtherMin = 30; //right:close, left:open
25
26 const int speedMax = 6;
27 const int speedMin = 0;
28 const int speedReverse = -3;
29 const int thSpeedReverse = 15; //0-100
```

【4】Minと付いた値が指が開いた角度、Maxと付いた値が指が閉じた角度を示しています。今回、開いているはずの状態でも指が曲がっていたので、より開くためにMinの値を少しずつ小さくしていきます。

今回の場合、

```
const int outIndexMin = 30;
```

だったものを

```
const int outIndexMin = 12;
```

にしてプログラムをアップロードすると、指が正しい角度になりました。

```
17  
18 const int outThumbMax = 150; //right:open, left:close  
19 const int outIndexMax = 150; //right:open, left:close  
20 const int outOtherMax = 150; //right:open, left:close  
21  
22 const int outThumbMin = 30; //right:close, left:open  
23 const int outIndexMin = 12; //right:close, left:open  
24 const int outOtherMin = 30; //right:close, left:open  
25
```



【5】次に親指の調整です。親指は開きすぎている状態なので、Minの値を少しずつ大きくしていきます。

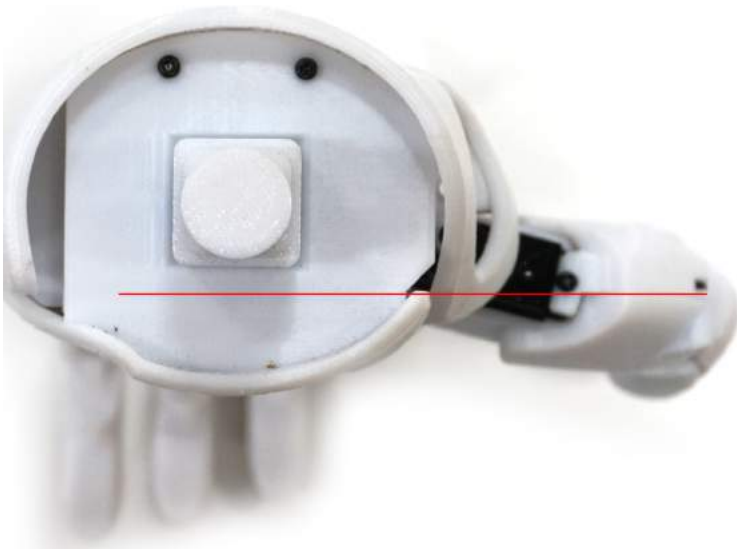
```
const int outThumbMin = 30;
```

だったものを

```
const int outThumbMin = 55;
```

にしてプログラムをアップロードした時、指が正しい角度になりました。

```
17  
18 const int outThumbMax = 150;//right:open, left:close  
19 const int outIndexMax = 150;//right:open, left:close  
20 const int outOtherMax = 150;//right:open, left:close  
21  
22 const int outThumbMin = 55;//right:close, left:open  
23 const int outIndexMin = 12;//right:close, left:open  
24 const int outOtherMin = 30;//right:close, left:open  
25
```



【6】次に指が閉じた状態の値を設定します。閉じた状態と開いた状態の角度差は、それぞれのモーターにおいて、次の値を参考にしてください。

```
-thumb: 86  
-index: 117  
-other: 55
```

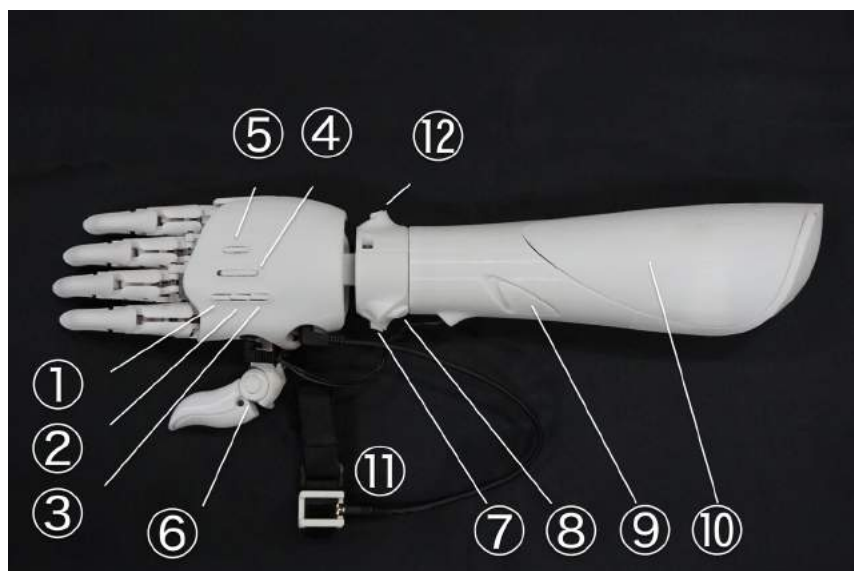
例えば、先程親指の開いた状態の角度を55と設定したので、55+86 と記入します。他の値も同様に設定しプログラムをアップロードします。

```
17  
18 const int outThumbMax = 55+86;//right:open, left:close  
19 const int outIndexMax = 12+117;//right:open, left:close  
20 const int outOtherMax = 30+55;//right:open, left:close  
21
```


3. 動かしてみよう

01 各部名称	P146
02 バッテリー取り付け	P147
03 キャリブレーション	P148
04 手の取り付け	P150
05 手首の撓尺屈	P151
06 手首の内外転	P151

01 各部名称



- | | | |
|-----------------|-------------|---------------|
| 1. キャリブレーションボタン | 2. 拡張用ボタン | 3. 親指回転ボタン |
| 4. 電源スイッチ | 5. 三指固定ボタン | 6. 親指ロックボタン |
| 7. 手首撓尺屈・掌背屈ボタン | 8. 手首回旋ボタン | 9. バッテリードアレバー |
| 10. バッテリードア | 11. 簡易圧力センサ | 12. 手首着脱ボタン |

How to Use HACKberry

02 バッテリー取り付け

バッテリードアレバーをスライドさせてバッテリードアを開き、バッテリーを装着します。

溝に沿ってスライドさせ、カチっという音がすることを確認してください。

(取り外すときはC-B-02を押しながら逆方向にスライドさせて下さい。)



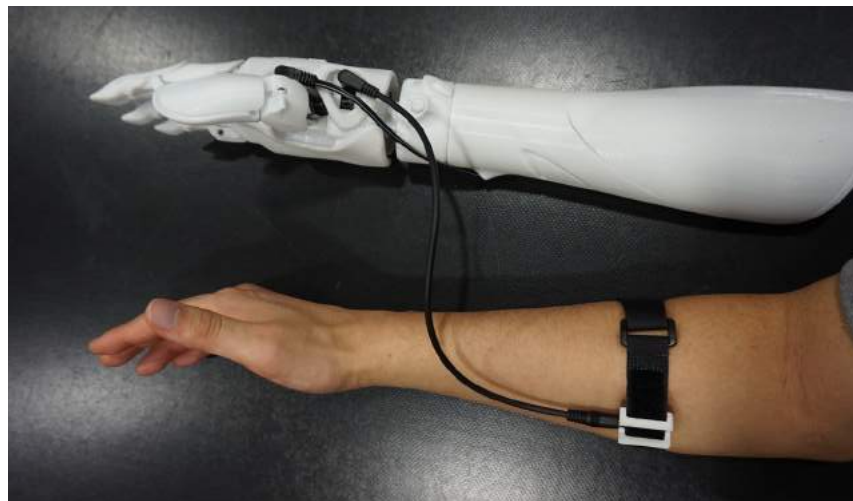
How to Use HACKberry

03 キャリブレーション

DCプラグを図のようにDCジャックに挿入します。



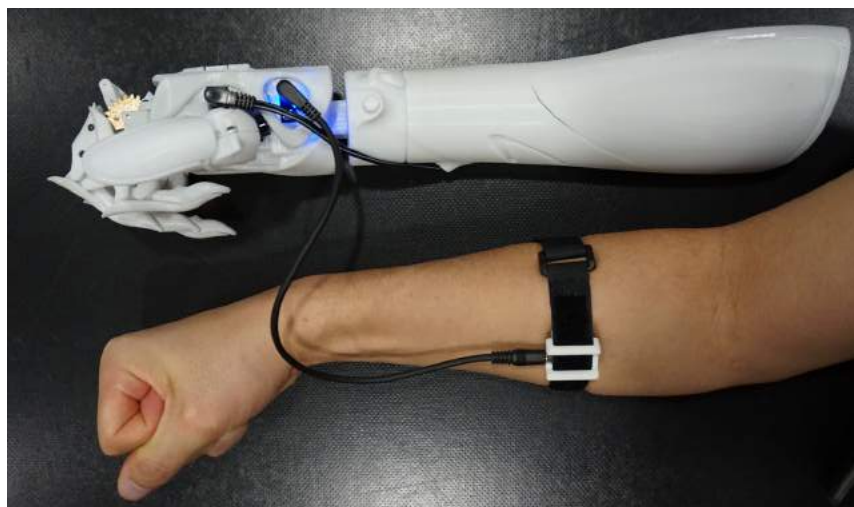
センサの平面部分を皮膚（なるべく筋肉のある部分）に当て、ベルクロストラップで軽く締め付けるように腕に固定します。その後、HACKberryとセンサをオーディオケーブルで接続します。**しっかりと奥まで差し込むよう注意して下さい。**



次にキャリブレーションを行います。

キャリブレーションではユーザーごとに異なる筋肉の隆起量をHACKberryに教えます。

電源が入った状態でキャリブレーションボタンを押し、4秒間の間、「拳を握る」→「力を抜く」という動作を数回繰り返して下さい。

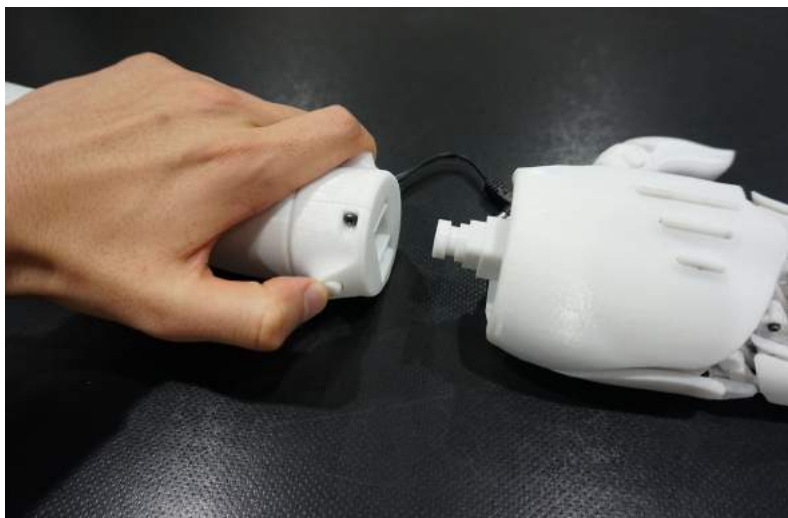


キャリブレーションが終わると、人差し指から小指までが揃って動きます。

How to Use HACKberry

04 手の着脱

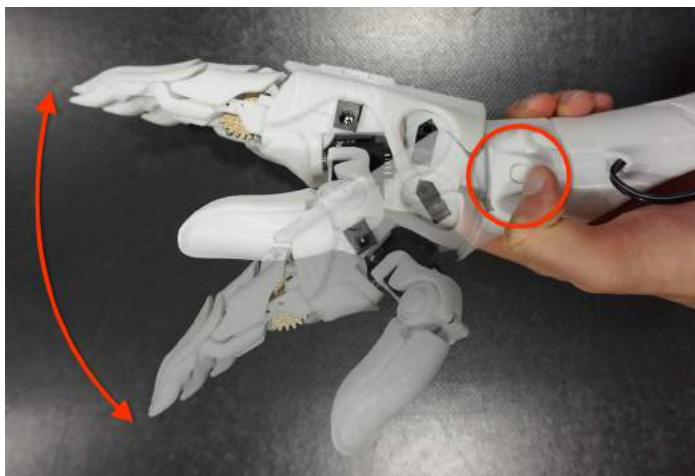
手部着脱ボタンを押すと、手部と手首の着脱を行えます。
手部と手首部の配置は90度ごとに調整できます。



How to Use HACKberry

05 手首の掌背屈・橈尺屈

手首橈尺屈・掌背屈ボタンを押すと、手首を掌背屈方向あるいは橈尺屈に傾けられます。



How to Use HACKberry

06 手首の内外転

手首内外転ボタンを押すと、手を図の方向に回転できます。



4.HACKberryをもっと楽しもう

- 01 はじめにP154
- 02 HACKberry Handboard Mk2 仕様P155
- 03 筋電センサによるHACKberryの操作P158
- 04 アセトン表面処理P162
- 05 ナイロン染色P168

本章ではHACKberryのカスタマイズ事例を紹介します。

1節では、Mk1基板に比べてカスタマイズ性が向上した
HACKberry hand board Mk2基板の特徴や仕様について説明します。

2節では、実際にMk2基板と市販の筋電センサ(2ch)を用いて、HACKberryを筋
電義手として操作する方法についてご紹介します。

3節では、家庭用の3Dプリント部品の見た目のクオリティを向上させる
アセトンの表面処理方法についてご紹介します。

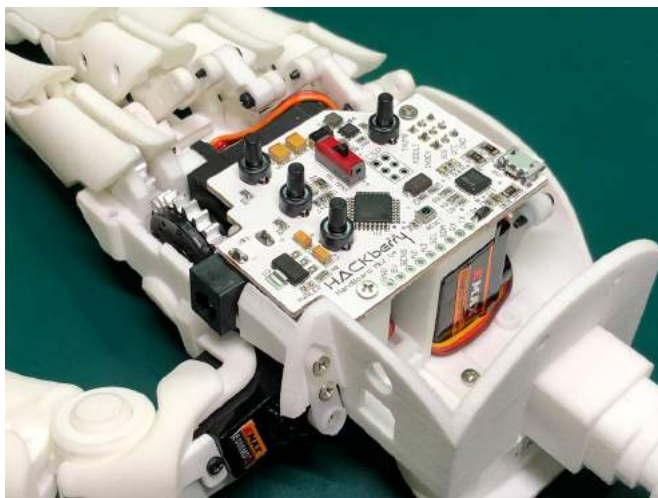
4節では、外部の3Dプリントサービスを利用して印刷した
ナイロン素材の3Dプリント部品を染色する方法についてご紹介します。

Customize

02 HACKberry hand board Mk2 仕様

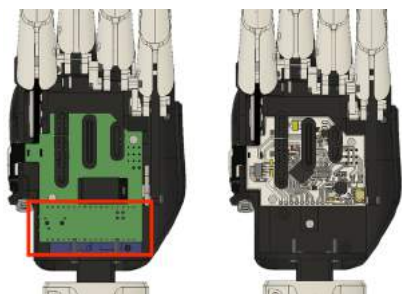
HACKberry hand board Mk2は、主にMk1からのカスタマイズ性の向上を目的として開発されました。次の二つが主な改良点です。

1. 基板一体化によるスペース確保
2. 拡張ポート増設によるカスタマイズの簡易化

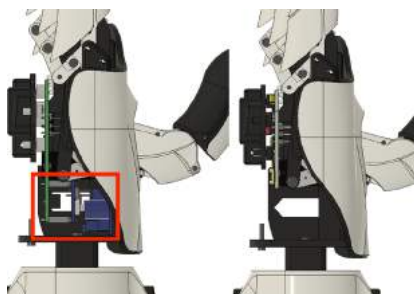


【1】基板一体化によるスペース確保

Mk1では分かれていたArduino MicroおよびDCDCコンバーターの機能を一枚の基板上に実装しました。これにより、筐体中の電気回路の専有スペースが減り、逆に、追加のセンサやモジュールを搭載するスペースが増えました。



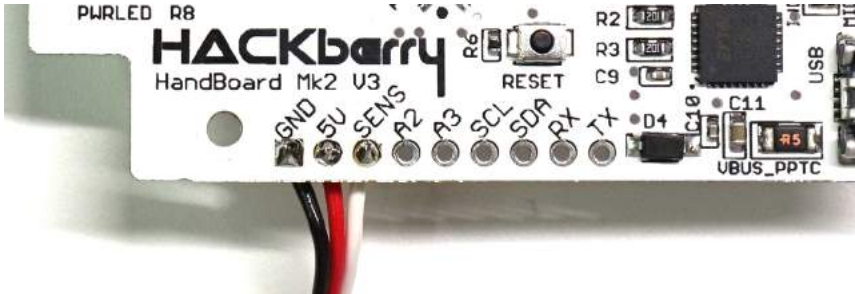
※スペース比較上面図(左 Mk1、右 Mk2)



※スペース比較側面図(左 Mk1、右 Mk2)

【2】拡張ポート増設によるカスタマイズの簡易化

基板下部にカスタマイズ用の入力ポートと通信ポート（アナログ入力、I2C通信、シリアル通信）を増設しました。よって、別のセンサやモジュールとの接続を簡単に行えるようになりました。



※基板下部のカスタマイズ用の入力・通信ポート

下に仕様の詳細を示します。Mk2基板はArduino互換機であり、プロセッサにはATmega328P-AUを採用しているため、ピンアサインはATmega328Pと同じです。

ピンアサイン	入力ピン	キャリブレーションボタン	A6
		拇指内外転ボタン	A0
		三指駆動ボタン	D10
		エクストラボタン	A7
		センサ入力	A1
	出力ピン	人差し指用サーボモータ	D5
		三指用サーボモータ	D6
		親指用サーボモータ	D9
	拡張ピン	5V	5V
		GND	GND
アナログ入力ピン		SENS(A1)	
		A2	
I2C通信ピン		A3	
		SCL	
シリアル通信ピン		SDA	
	RX		
		TX	
書き込み時の設定	ボード	Arduino Nano	
	プロセッサ	ATmega328P-AU	
	書き込み装置	AVRISP mk II	
入力電圧	推奨入力電圧	7V~12V	
	最大入力電圧	20V	
	DCDCコンバータ定格入力電圧12V, 最大24V		
	三端子レギュレーター最大入力電圧20V より		
サーボピン	PPTCはthumbには入っていません		
	IndexとMiddleは500mAが限度です。		

※ HACKberry hand board Mk2 詳細仕様

03 筋電センサによるHACKberryの操作

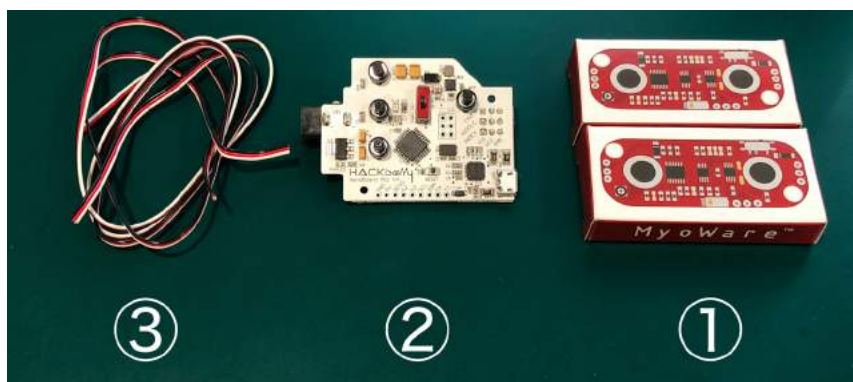
【1】はじめに

市販の多くの義手では筋電センサが使われているのに対して、HACKberryでは簡易圧力センサを使用しています。簡易圧力センサの特長は、低ノイズで操作しやすく、極めて低価格なことです。一方で、多チャンネル化が難しく1チャンネルの単純な信号しか測定出来ないというデメリットもあります。HACKberryを市販の義手同様に筋電センサを用いて操作したい場合は、本節を参考にカスタマイズしてください。

【2】筋電(Myoelectricity)とは

人が筋肉を動かす時、脳から発せられた電気信号が神経を經由して筋肉に伝えられます。この電気信号は皮膚表面に僅かに漏れ出てきます。これを皮膚表面に貼った電極により観測した信号が筋電です。

【3】使用部品



1. MyoWare 2個

Advancer Technologies社製の筋電センサ。筋電位の増幅や平滑化回路を内蔵し、比較的簡単に筋電信号を計測できます。

2. HACKberry hand board Mk2

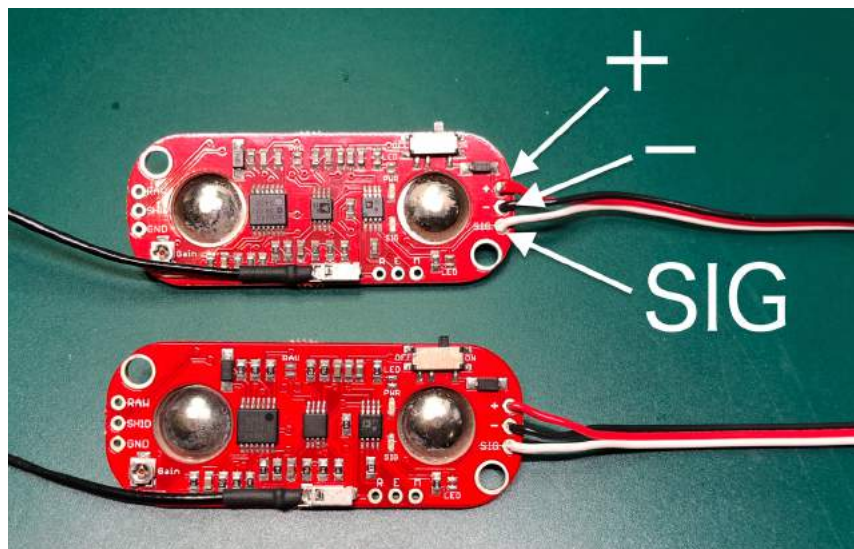
オーディオジャックは接続しないで用います。

3. 3列フラットケーブル 50cm 2本

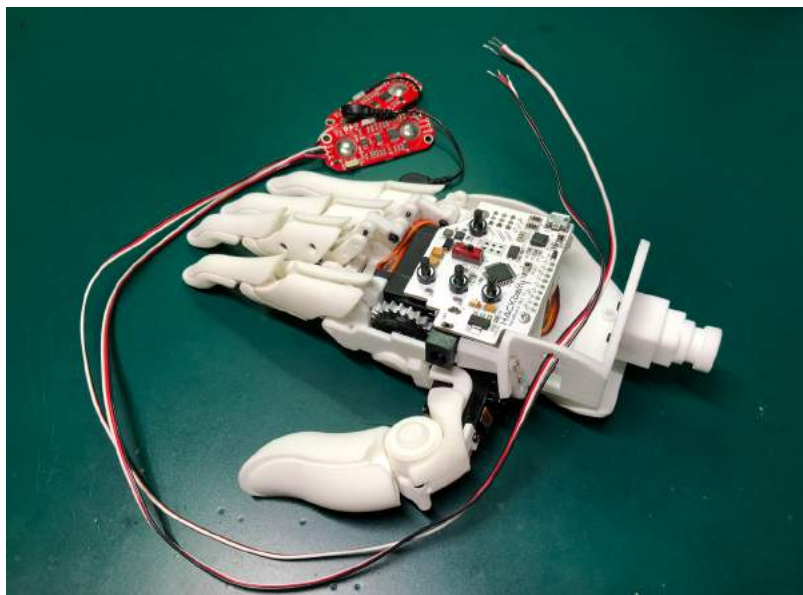
本節では、MyoWareを2つ使用し、それぞれ腕の掌屈（掌側に手首を曲げる）および背屈（手の甲側に手首を曲げる）の運動を検出して、2チャンネルによるHACKberryの操作を実現します。

【4】実装方法

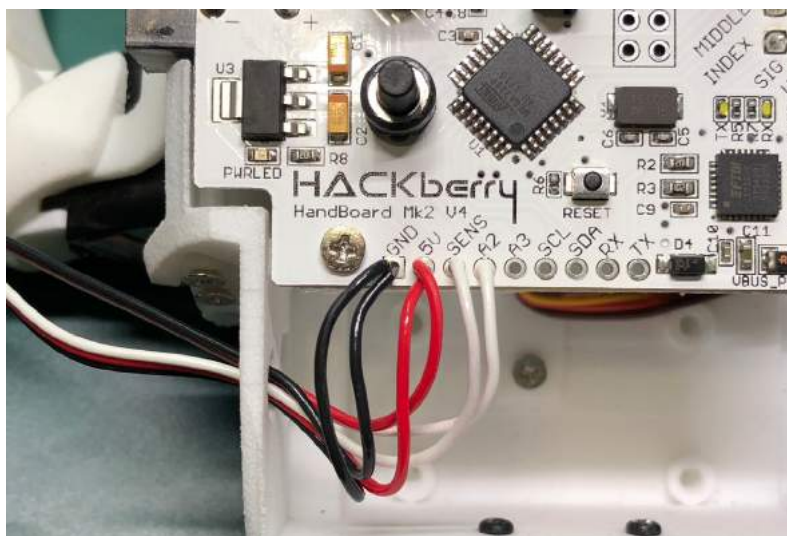
MyoWareの+、-、SIGの3箇所それぞれ赤、黒、白の線をはんだ付けします。



HACKberryにHACKberry hand board Mk2を取り付け、図のようにフラットケーブルをオーディオジャックの穴に通します。



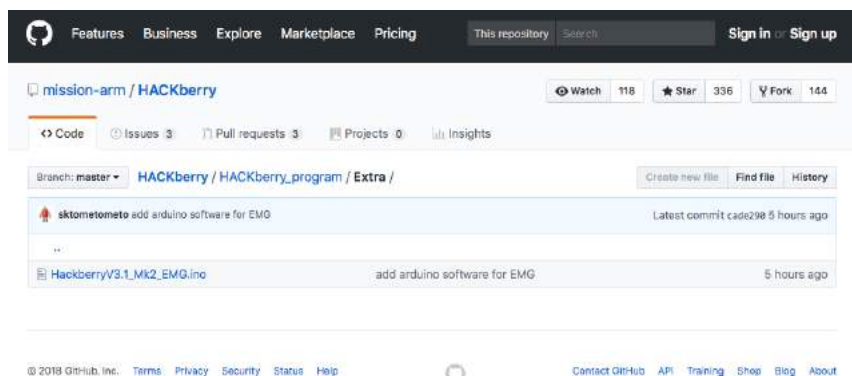
HACKberry hand board Mk2の拡張ポートにフラットケーブルをはんだ付けします。図のように、2本の黒い線をGNDに、2本の赤い線を5Vに、一方の白い線をSENSに、もう一方の白い線をA2に接続します。



これでハードウェアの作業は終了です。

次に筋電操作をするためのプログラムをGithubからダウンロードします。

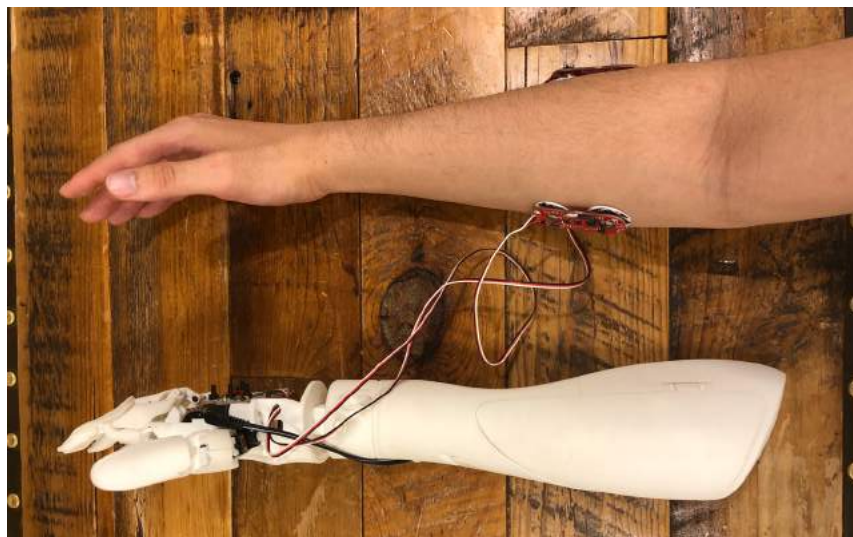
[02 部品の入手方法] (19ページ)を参考にしてHACKberryのGithubページにアクセスし、Program配下にある“HackberryV3.1_Mk2_EMG_180412”をダウンロードしてください。



その後、[05 プログラムのアップロード] (52ページ)を参考にして先程ダウンロードしたプログラムをHACKberry hand board Mk2に書き込みます。

【5】操作方法

白いコードがHACKberry hand board Mk2 外部ポートのSENSに接続されているMyoWareを屈筋（掌屈の際に盛り上がる筋肉）に、もう片方のMyoWareを伸筋（背屈の際に盛り上がる筋肉）に取り付けます。詳しいセンサの取り付け位置はMyoWareの取扱説明書を参考にしてください。



HACKberryの電源を入れ、キャリブレーションボタンを押します。ボタンを押してから4秒間キャリブレーションが行われるので、その間に屈筋と伸筋に交互に力を入れて、筋電信号の幅をHACKberryにプリセットします。キャリブレーション終了後は屈筋に力を入れれば指が屈曲し、伸筋に力を入れれば指が伸展します。



YouTube筋電操作動画

<https://youtu.be/pDvJqjNldBo>

03 気化アセトンによる表面処理

[1]はじめに

家庭用3Dプリンタで一般的に用いられているFDM(熱溶解積層法)は安価に立体形状を印刷することができますが、一層ずつ積み重ねるといったその性質上、表面の層の粗さが目立ってしまいます。その場合、気化アセトンによる表面処理をすることにより、表面を滑らかにすることができます。この手法のメリットとデメリットを以下に挙げます。

メリット

- 所要時間が短い(紙やすりなどを用いて表面を削り滑らかにする方法と比べておよそ1/10程度。)
- ヤスリがけでは難しい細かい部分も処理可能。
- 3Dプリント時の積層ピッチが大きくても効果があるので、プリント時間を大きく削減できる。
- 表面を溶かして均(なら)すので、ある種の熔融状態になり、層間の結合力の弱さが軽減され、全体的に強度が向上する。

デメリット

- 有機溶剤を使うため、使用に注意が必要。
- 全体的に表面を溶かして滑らかにするという特性上、ディテールが潰れる可能性があり、精度の必要な嵌め合い、圧入には向かない。

これらデメリットへの対策としては、筐体の表面に露出する部分をその他の構造・機構部分から分けてこれに対してのみ表面処理をしたり、表面処理したい部位をアセトンに当たりやすい位置(後述)に置いたり、などの方法があります。

[2]注意点

アセトンは有機溶剤です。消防法により危険物第四類(第一石油類 危険等級2 水溶性)に指定されています。特に注意すべきアセトンの特徴は、「**強い引火性(常温で引火)**」と「**高い揮発性**」です。以下の点に注意してください。

- 引火や過度の吸入を防ぐため、**通気・換気の良い場所で取り扱い、蒸気をためこまないように**しましょう。
- 作業中に容器からこぼれた場合は、素早く布でふき取りましょう。
- 目にアセトンの蒸気が入った場合は大量の水で素早く洗いましょう。たとえ、痛みや異常がなかったとしてもすぐ専門医療施設で診察を受けることをお勧めします。
- 誤って飲みこんだり、ガスを吸いこんだりしたときも医師の診察を受けることをお勧めします。
- アセトンを使用した後は**手洗いうがい**を徹底するように心がけてください。

【3】機材

- 密閉容器（アセトンで溶けない素材。ここでは 5 L のポリエチレン容器を使用。）
- アルミホイル
- キッチンペーパー
- アセトン（ホートク 純アセトン1L <http://amzn.asia/4U79i9I>）
- ピペット

【4】手順

【1】密閉容器の底面にアルミホイルを敷きます。アルミホイルは容器の外側にはみ出るように大きめに切り取ります。これは、後ほど容器から部品を取り出す時に、アルミホイルの縁を持って部品に触れないようにするためです。



【2】敷いたアルミホイルの上に部品を並べます。この時、**表面処理したい面を上側にします**。これは気化したアセトンが上から部品に降りかかる為、上部の方がより溶けやすいからです。



【3】キッチンペーパーを3枚、2つ折りにして写真のようにセットし、ピペットを用いて1.5 ml のアセトンを含ませます。





【4】その後すぐに蓋をしてタイマーをスタートさせます。0.3mmピッチで印刷したものならば1時間程度を目安としてください。



【5】所定の時間が経った後に、蓋を開けて部品を取り出します。この時、部品は柔らかくなっているので、**部品には直接触れずにアルミホイルの縁を掴んで取り出します。**



【6】表面が溶け、滑らかになっているのがわかります。このまま**4～5時間風通しの良い場所に放置**します。



【7】表面処理が目的ならばこの時点でおおよそ滑らかになっていますが、テカリが気になるという場合はつや消しスプレーなどを吹くことで元と同じような質感に戻ります。



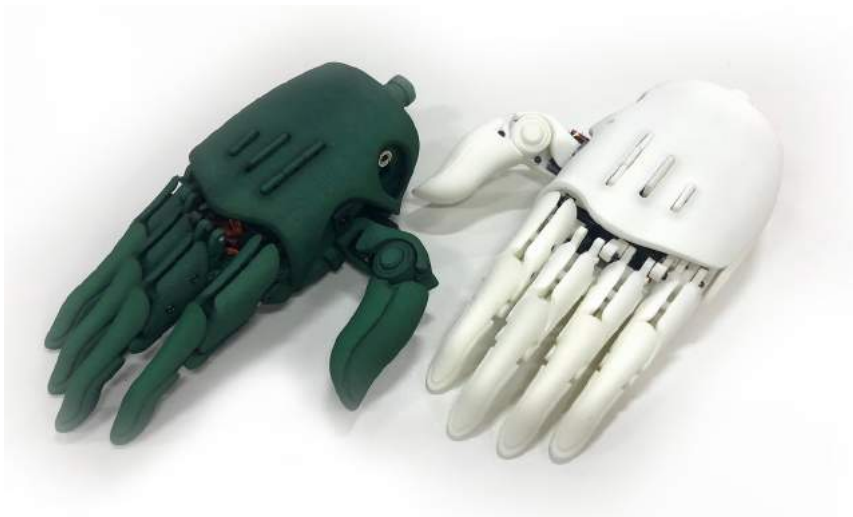
※つや消し前の図



※つや消し後の図

Customize

04 ナイロン染色



【1】はじめに

本節では、HACKberryを着色する方法の一つとして、ナイロン素材で3Dプリントした部品を市販の樹脂用染料を用いて染色します。

ナイロン素材の3Dプリント部品は白く出力されるため染色をしやすいです。(それだけでなく、ナイロン素材は低価格かつ高強度であるため、我々はHACKberryの素材として標準的に採用しています。)

ナイロンは家庭用のFDM(熱溶解積層法)方式の3Dプリンタには対応していないのでご注意ください。(家庭用の3Dプリンタの樹脂には、一般的にABSやPLAを用います。ABSやPLAは染色に不向きです。)

ナイロンはSLS(粉末焼結型)方式という粉末を一層一層レーザーなどで焼き固めて積層する方式の3Dプリンタに対応しています。SLS方式の3Dプリンタは家庭用には普及していないため、外部の3Dプリントサービスを利用することをおすすめします(外部のサービスを使うとナイロン以外にも様々な素材を選べます)。我々は、DMM.comの3Dプリントサービスを使用しています。

DMM.com 3Dプリントサービス：<https://make.dmm.com/print/>

ナイロンを取り扱っている3Dプリントサービスでは、同時に染色のサービスも提供している場合があります。ここでは、好きな色を好みの染め具合に調節したいので、市販されている樹脂用の染料を用いて自分で染色をしていきます。

【2】機材



1. 樹脂用染料を希釈した染め液
2. ステンレスザル
3. キッチンペーパー
4. ゴム手袋
5. 計量カップ
6. 漏斗
7. 菜箸
8. 温度計
9. IHヒーター (染め液の加熱用なのでガスコンロなどでも可)
10. ステンレス鍋 (ザルが入るサイズ)

【3】染色の流れ

【1】始めに染め液を作ります。今回は大阪化成品株式会社の「樹脂用染料SDN」を使用しました。染めたい部品が浸かる位の量を目安にして計量カップを使って染料を20倍に希釈します。染料によって希釈の濃度が違うので注意してください。



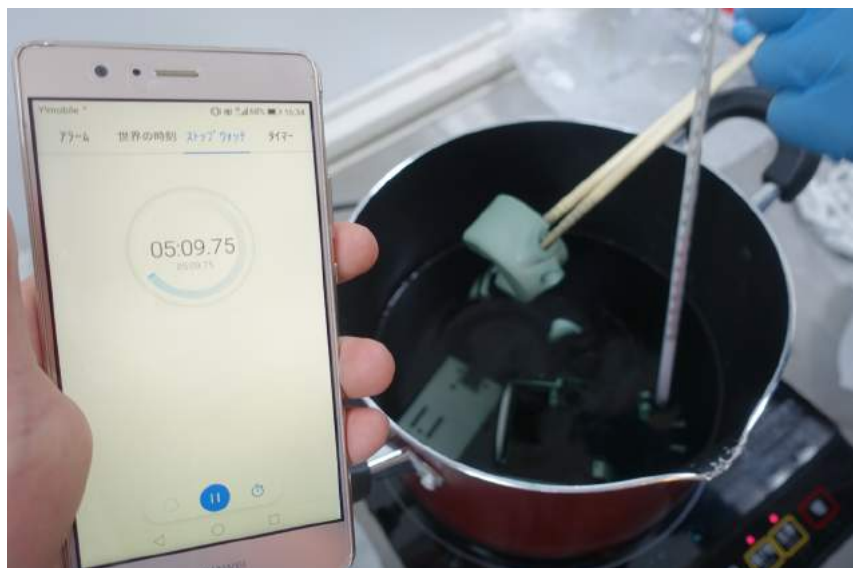
【2】ステンレス鍋に染め液を入れて加熱します。



【3】「樹脂染料SDN」のナイロンにおける標準染色条件は $70 \pm 10^{\circ}\text{C}$ で3~5分です。染め液の温度が $70 \pm 10^{\circ}\text{C}$ の範囲まで上がったら3Dプリント造形物を染め液に浸しタイマーをスタートさせます。染色している間は指定温度の $70 \pm 10^{\circ}\text{C}$ に保ち続けます。また、適度に染め液を攪拌します。



【4】メーカーの提示する染色時間は3～5分とのことですが、我々の環境では5分後に引き上げてもあまり染まっていないようでしたので、20分程染色を行いました。



※ 5分後



※ 20分後

【5】好みの色に染まったら染め液から3Dプリント造形物を取り出してキッチンペーパーなどに並べて乾燥させます。これでナイロン素材の3Dプリント造形物の染色は完了です。



【6】染め液は人体にとって有害な物質を含んでいますので染色が終わっても**そのまま流しに捨てることは出来ません**。染め液は使い回しが出来るので漏斗を使ってペットボトルなどに保存しておきます。処分したい場合は、染め液を更に200倍以上に希釈して流しなどに捨てるようにします。



5. 故障かな？とおもったら

Troubleshooting

故障かな?と思ったら

電源が入らない

- ◆バッテリーが正しく取り付けられているか確認してください。
- ◆DCプラグがしっかりと基板のDCジャックに挿し込まれているか確認してください。
- ◆はんだ付けした位置が正しいか確認して下さい。
 - DCプラグの線は正しくプラス側とマイナス側にはんだ付けされていますか。
 - 基板上のVCC(またはV+)とGNDがショートしていませんか。
- ◆バッテリーの端子にテスターを当てて電圧が正しく表示されるか確認してください。
 - 回路がショートしていた場合、バッテリー内の保護回路が働き、電圧が発生しない状態になっている可能性があります。復旧方法に関しては各バッテリーメーカーにお問い合わせ下さい。

キャリブレーションが出来ない

- ◆キャリブレーションボタンを押した時、クリック感があるか確認して下さい。
 - ボタンが傾いてはんだ付けされたなどの理由から、ボタンが押されたままの状態になっている可能性があります。はんだ付けをやり直すか、もしくは筐体パーツを削って修復してください。
- ◆正しいバージョン(左か右か、mk1かmk2か)のプログラムを書き込んだか確認して下さい。
- ◆サーボモーターの初期位置設定が正しかったか確認してください(詳しくは134ページを参照してください。)
- ◆サーボモーターのピンソケットが正しい向きで基板に接続されているか確認して下さい。(詳しくは112ページを参照してください。)
- ◆センサが正しく接続されているか確認して下さい。
 - オーディオケーブルはしっかりと奥まで差し込まれていますか。
 - 正しく接続しても反応がない場合、センサかオーディオケーブル自体が破損している可能性があります。

指がしっかり曲がらない

- ◆サーボモーターの初期位置設定が正しかったか確認してください。(詳しくは134ページを参照してください。)

サーボモーターから異音がる

- ◆サーボモーターの初期位置設定が正しかったか確認してください。サーボモーターが構造的に回転出来ない方向に動かそうとすると異音が発生します。(詳しくは134ページを参照してください。)
- ◆DCDCコンバーターの電圧調整が正しかったか確認して下さい。電圧調整をしていないとサーボモーターに過剰な電圧がかかりサーボモーターが破損します。(詳しくは92ページを参照してください。)

突然サーボモーターが動かなくなった

- ◆(mk1の場合)ポリスイッチが正しく挿入されているか確認して下さい。
 - ➔挿入していない場合、サーボモーターが発熱し破損した恐れがあります。その場合、モーターを交換し、ポリスイッチを挿入して再度組み立てて下さい。
 - ➔挿入していても、はんだ付けの不良や絶縁不足でサーボモーター内の回路がショートしてしまった可能性があります。その場合はモーターを交換してください。
- ◆DCDCコンバーターの電圧調整が正しかったか確認して下さい。電圧調整をしていないとサーボモーターに過剰な電圧がかかりサーボモーターが破損します。(詳しくは92ページを参照してください。)

センサを押しても反応しない

- ◆サーボモーターの初期位置設定が正しかったか確認してください。(詳しくは134ページを参照してください。)
- ◆センサが正しく接続されているか確認して下さい。
 - ➔オーディオケーブルはしっかりと奥まで差し込まれていますか。
 - ➔正しく接続しても反応がない場合、センサかオーディオケーブル自体が破損している可能性があります。
- ◆センサを取り付ける位置を調整してみてください。
 - ➔なるべく筋が隆起しやすい位置に取り付けてください。

プログラムが書き込めない

- ◆通信機能のあるUSBケーブルを使用しているか確認して下さい。(充電専用のUSBケーブルでは通信できません。)
- ◆COMポートが正しく認識されているか確認して下さい。(54ページを参照して下さい。)
- ◆正しいボードを選択しているか確認してください。(56ページを参照して下さい。)
- ◆ドライバが正しくインストールされているか確認して下さい。
 - ➔ドライバは基本的には自動インストールされますが、環境によってはされない場合があります。ドライバのインストールは次のサイトを参考にしてください。
https://synapse.kyoto/tips/FTDI_driver/page001.html

指が暴走する

- ◆センサが正しく接続されているか確認して下さい。
 - ➔オーディオケーブルはしっかりと奥まで差し込まれていますか。
 - ➔正しく接続しても反応がない場合、センサかオーディオケーブル自体が破損している可能性があります。

HACKberry HANDBOOK

— 制 作 —

特定非営利活動法人 Mission ARM Japan

E-mail : info@mission-arm.jp

Web : <http://www.mission-arm.jp/>

HACKberry : <http://exiii-hackberry.com/>

Facebook : <https://www.facebook.com/mission.arm.jp>

ディレクター 近藤玄大

エディター 小笠原佑樹

表紙デザイン 中田邦彦

本文デザイン 後藤馨斗

本書の内容に関するお質問については、
Mission ARM japan ホームページ お問い合わせフォームにお送りください。



発行元
特定非営利活動法人 Mission ARM Japan
info@mission-arm.jp
第1版 2018年6月

<http://exiii-hackberry.com/>