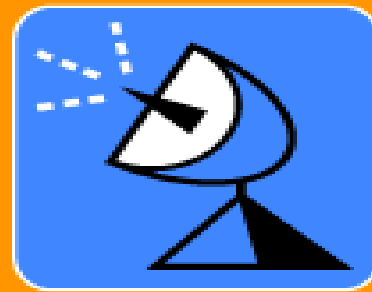




# 秋月AE-TB67H450とM5Stack(Basic)とRaspberry Pi を使った電磁石の制御その2(IoTLT版)

IoT縛りの勉強会! IoTLT vol.113 2024 7/17

福山大学工学部情報工学科 山之上卓



# 自己紹介



福山大学の教員  
高専OB

着る電光掲示板、  
テレポードレッサー、  
シン3次元表示装置

テレポーターション  
システム作成中

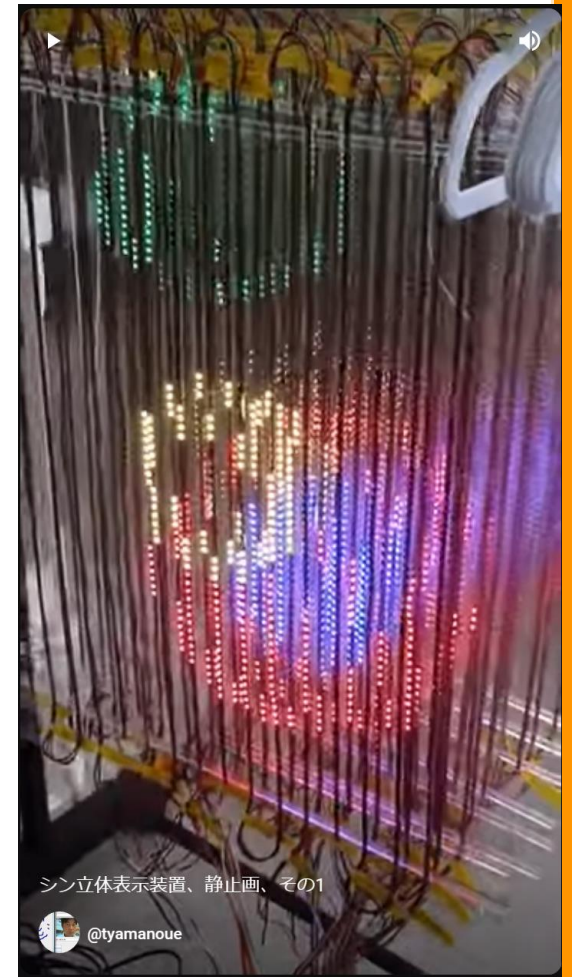




# 自己紹介



- 1980年 interface LKIT-16によるFORTRANシステム
- 1980年 bit ナノピコ教室じゃんけん大会優勝
- 1988年日本経済新聞「パソコン通信使い電算機言語開発」
- 2016年 ACM SIGUCCS Hall of Fame
- 「ギャル電の意識の低いプレゼンバトル」優勝  
@Geek night 広島, 2017 12/9
- MF Tokyo 2018, MF Taipei 2018, NT広島2019
- NT金沢2022, OMMF2022, MFT2023, MFK2024
- ヒーローズ・リーグ2023 トランジスタ技術賞受賞
- 情報処理学会フェロー

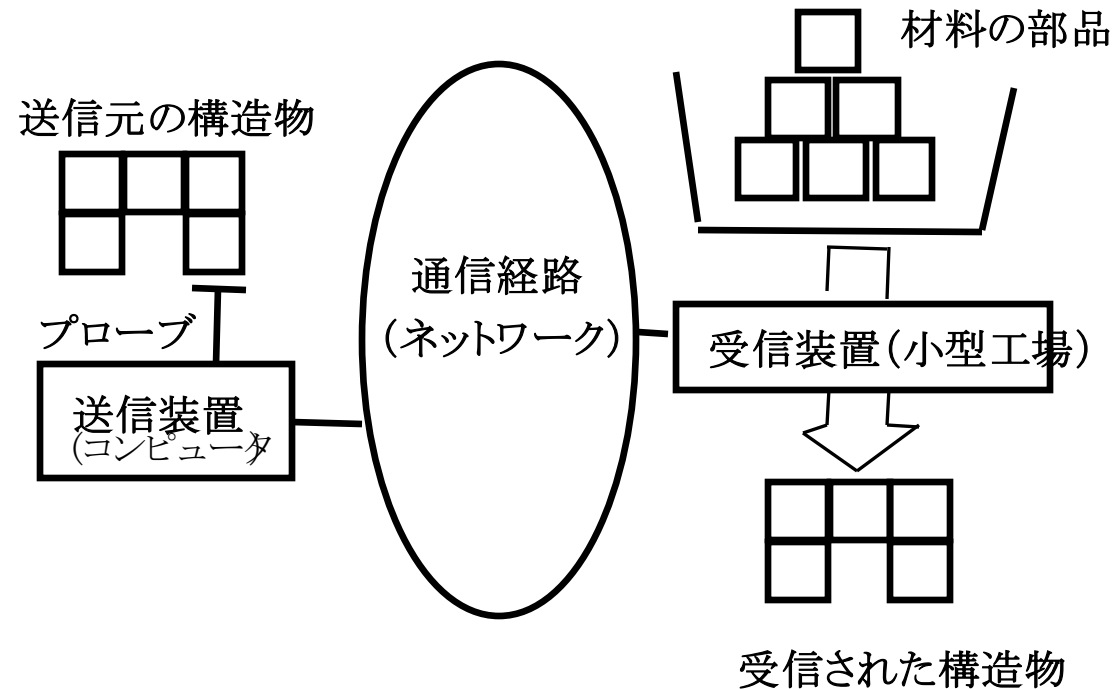


シン立体表示装置、静止画、その1

# 背景



- ・ テレポーテーション (物体瞬間移動) を実現したい！





## 経緯

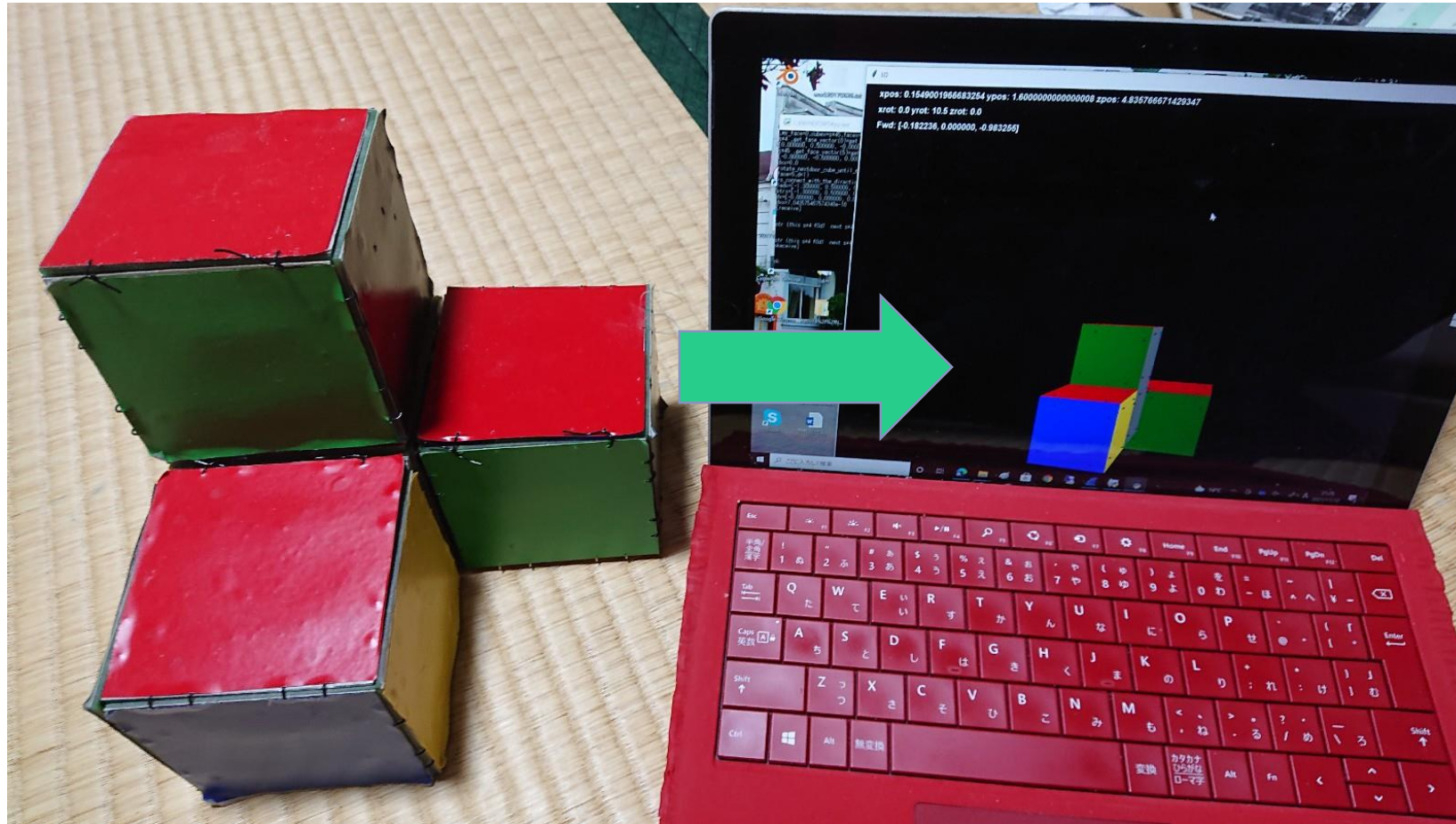


- ・ 1994 うどんをネットワークで送りたい@JAINコンソーシアム@志賀島
- ・ 1995年、What How For 北九州(産学官交流サロン)で開発スタート
- ・ 1999年、特許出願(2009年査定)
- ・ 2002年、試作開始、シミュレータ作成
- ・ 2021年、Real→Virtual 転送成功(IoTLT vol.93)



Project  
TESLA

# 背景その2 Real→Virtual





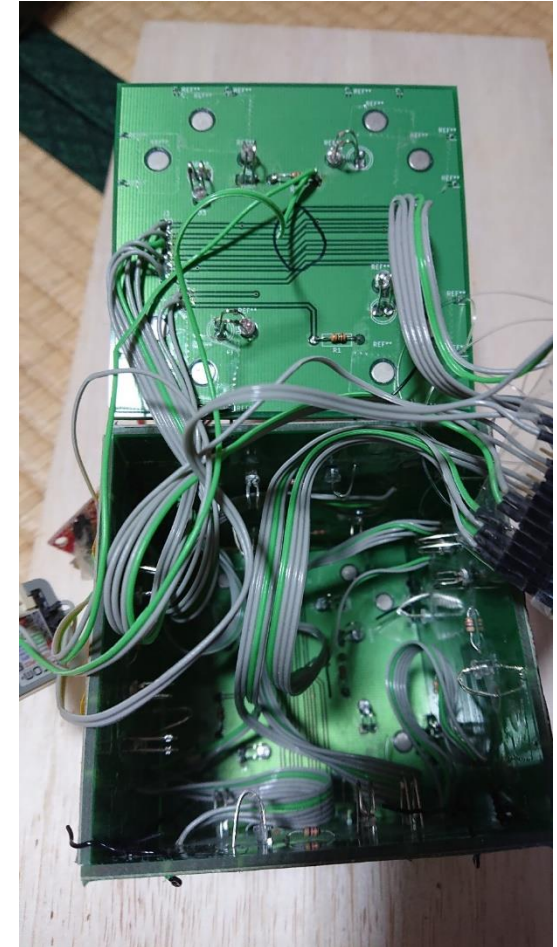
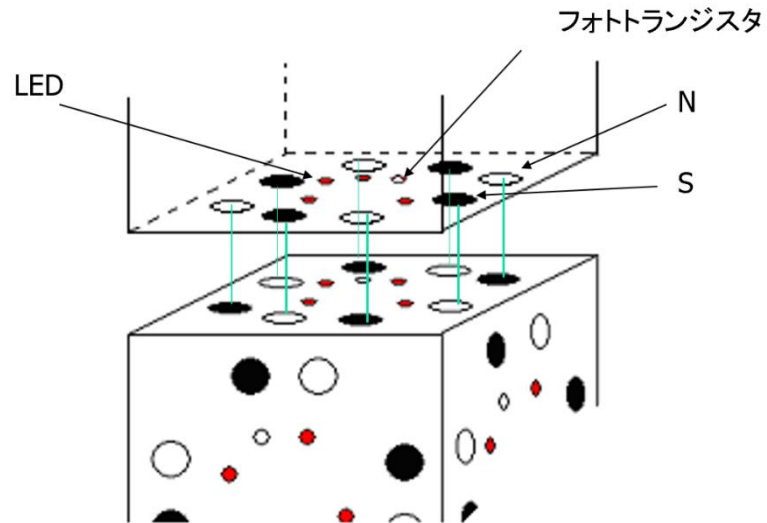
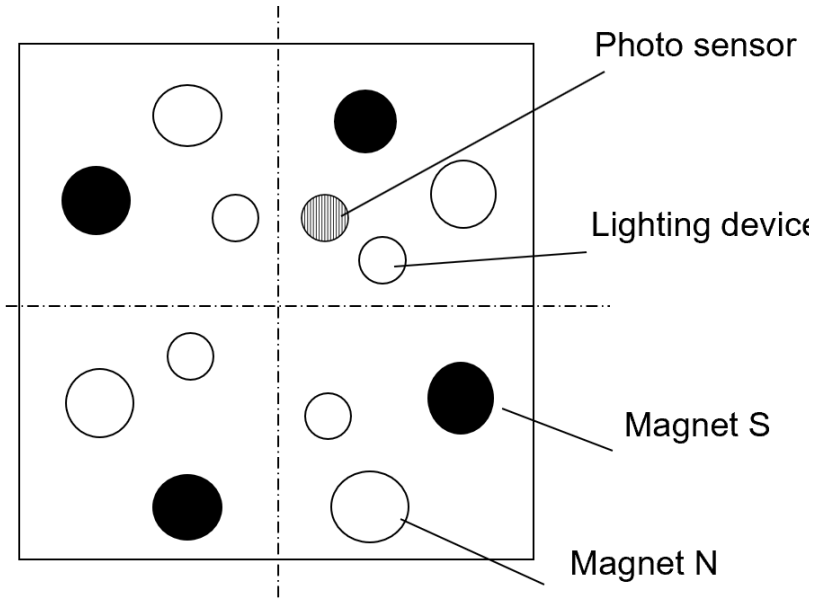
## 経緯



- ・ 2023年12月 Qiita, IoTLT Advent Calendar 11日目  
「秋月AE-TB67H450とM5Stack(Basic)とRaspberry Piを使った電磁石の制御」
  - 電磁石を使ったテスラダイスの把握と分離
    - ・ ネオジム磁石の脅威
      - 電流を流さないのにくっつく
      - 電流の向きを変えても離れない



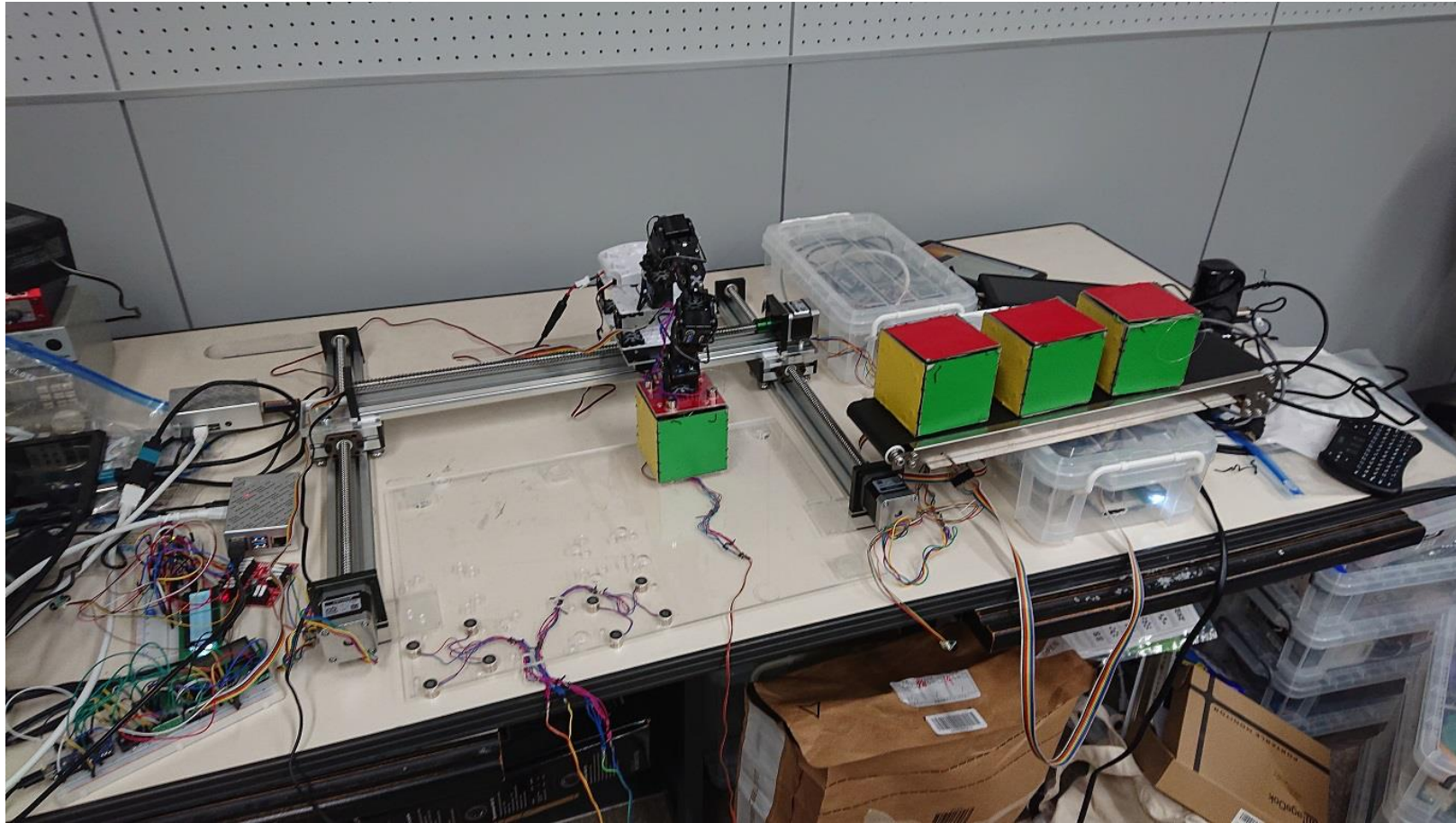
# 部品間接続・通信機構





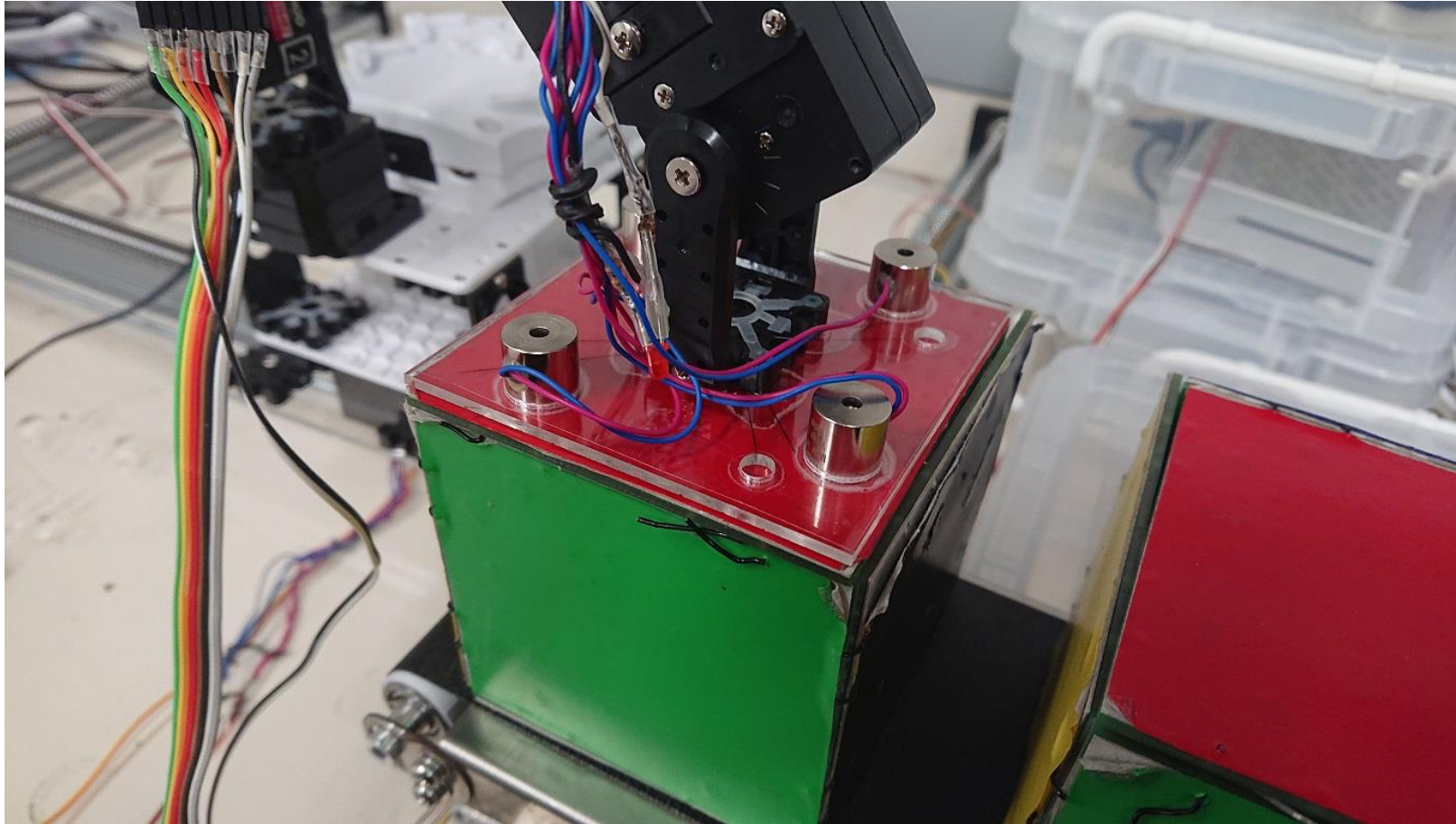


# 製作中の受信装置(自動工場)





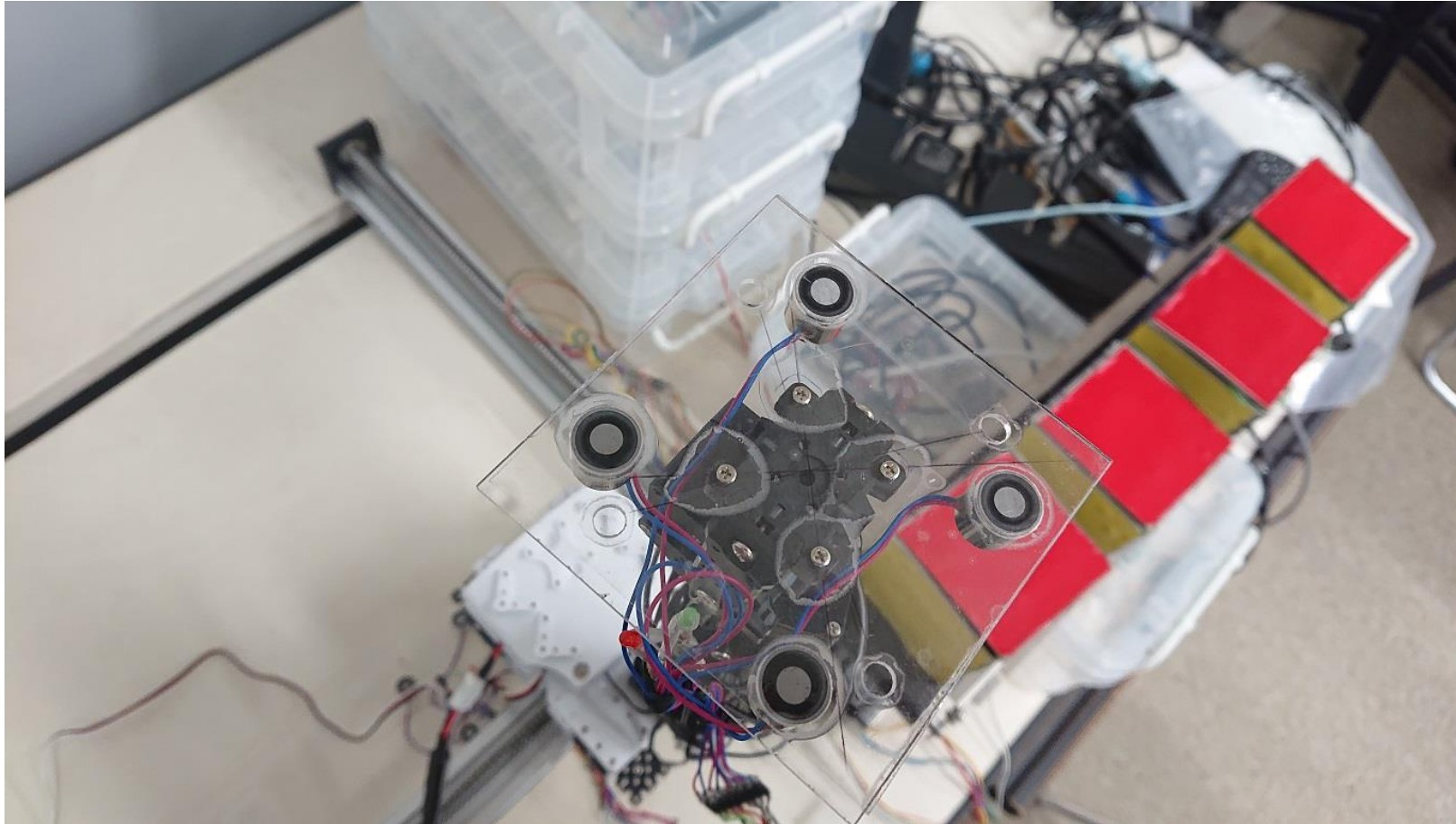
# ロボットアームの 部品把握部分





Project  
TESLA

# ロボットアームの 部品把握部分





やってみた

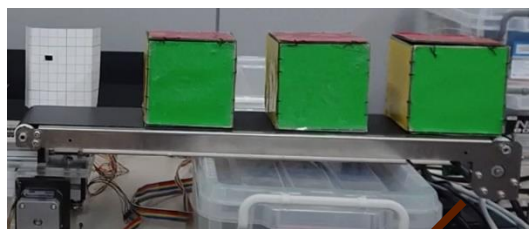


- ・ うまくいかないかもしれないけど、とにかくやってみた
  - Youtube : <https://youtu.be/JHxr8vVwPaU?si=Qs8htKyiGSgnQcD3>

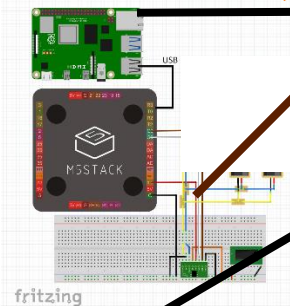




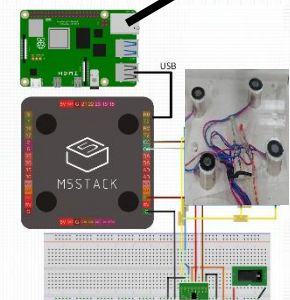
# Hardware



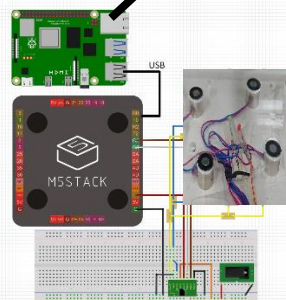
ベルトコンベアとその制御



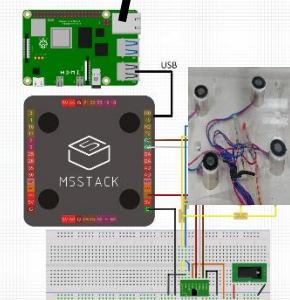
fritzing



fritzing

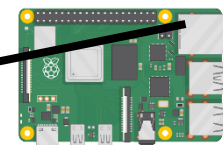
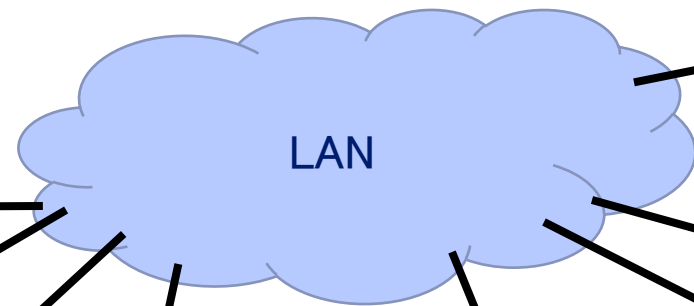


fritzing

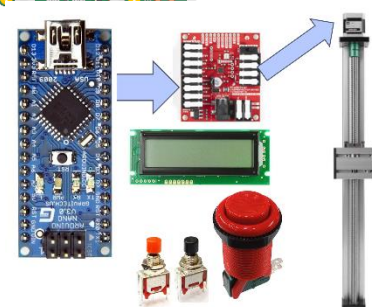
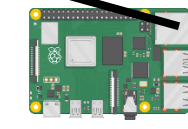
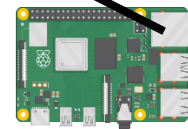
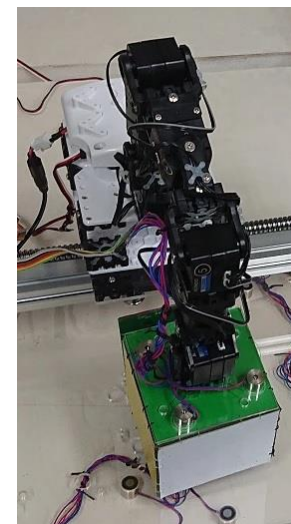


fritzing

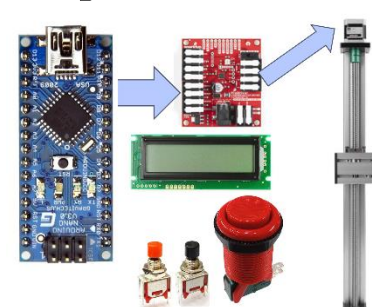
電磁石とその制御



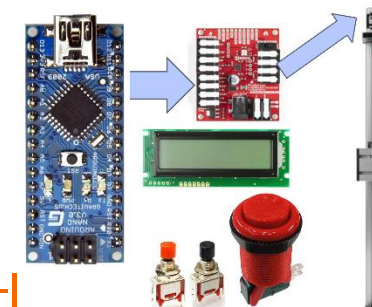
RCB-4mini  
ロボットアームとその制御



x



y-l



y-r

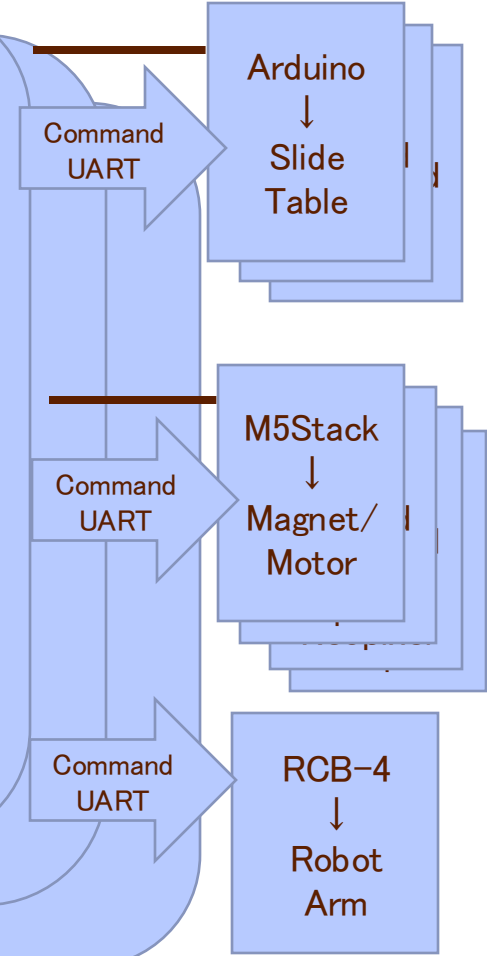
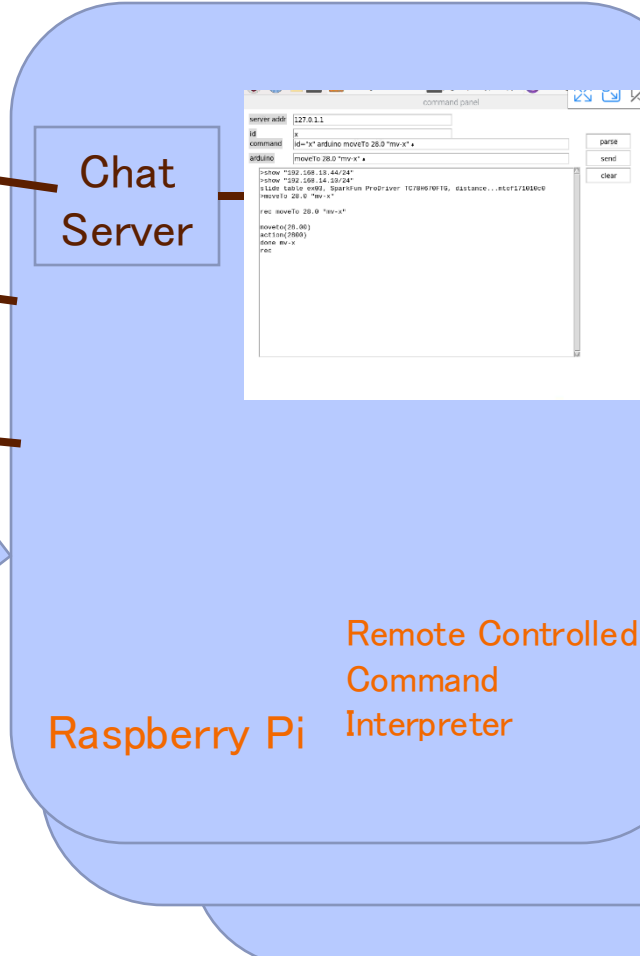
ダイス(部品)の水平方向の移動



# Software



```
command panel
main script
Servers connect set id set id
command enter run script
command for ith 0 enter
command for all enter clear_message
connected to 192.168.1.34:9998
send set id="x-l" to 0
send id="x-[a-z]+" get distance to 0
send id="x-[a-z]" arduino get distance to 0
from 192.168.1.34:9998, [receive]- rec get distance
from 192.168.1.34:9998, [receive]-
from 192.168.1.34:9998, [receive]- 15 mm.
send id="x-l" arduino get distance to 0
from 192.168.1.34:9998, [receive]- rec get distance
from 192.168.1.34:9998, [receive]- 15 mm.
send id="y-l" arduino get distance to 0
```



Main Controller  
Command Interpreter, Communication(TCP client),  
PyRealSense, ...



# ロボットアームの制御



## ロボットアーム

- Kondo KHR-A5, サーボモータ追加。高出力サーボモータに一部交換

## アームの動きの作成:

- Windowsの Heart2Heart4 で作成, RCB-4mini に書き込み  
TEACH 機能大活躍
- Raspberry Pi -動作番号->UART->RCB-4mini
- Heart2Heart4, RCB-4 Library for Python
  - ・ Raspberry Pi<->RCB-4mini

The screenshot displays the Heart2Heart4 software interface. The main window shows a motion sequence for 'motion-pick-01' with positions Pos0, Pos1, and Pos2. A 'Position' dialog box is open, showing various coordinate values for 'ハンド' (Hand) and '先端' (Tip) in 'ヨー' (Yaw) and 'ピッチ' (Pitch) directions. The 'Motion Table' on the right lists 25 motion steps (M001 to M025) with their names, button numbers, and comparison symbols.

番号	名前	ボタン番号	比較
M001	motion-pick-01	B:0	=
M002	motion-up-01	B:0	=
M003	motion-place-m1	B:0	=
M004	motion-release-m1	B:0	=
M005	motion-to-home	B:0	=
M006	motion-place-2	B:0	=
M007	motion-grasp-m1-01	B:0	=
M008	motion-rotate-m1-01	B:0	=
M009	motion-rotate-m1-02	B:0	=
M010	XA5_291 ホームポジション	B:0	=
M011	XA5_292 電圧低下	B:0	=
M012	motion-grasp-m1-02	B:0	=
M013	motion-rotate-hc-01	B:0	=
M014	motion-rotate-hc-02	B:0	=
M015	motion-release-c-02	B:0	=
M016	motion-release-m1-03	B:0	=
M017	motion-grasp-m1-c-01	B:0	=
M018	motion-rotate-hc-03	B:0	=
M019	M019	B:0	=
M020	M020	B:0	=
M021	M021	B:0	=
M022	M022	B:0	=
M023	M023	B:0	=
M024	M024	B:0	=
M025	M025	B:0	=



## 全体の制御



- 受信装置の制御は、main controllerでpython の文の列をexec関数で実行することで実施。
- 各文は、スライドテーブル、ベルトコンベア、電磁石、アームを制御する機構のついたRaspberry Pi+(Arduino/M5Stack/RCB4-mini)へコマンドを送り、各担当機器を制御。

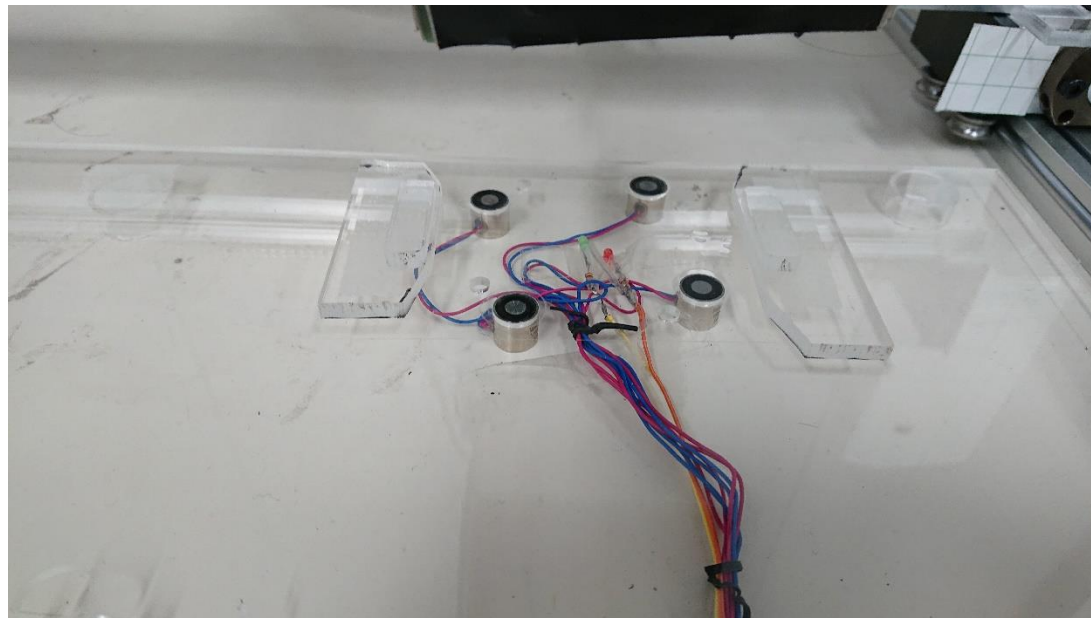
```
time.sleep(10.0)
self.send_init_belt_command()
self.send_arm_motion_command("1")
time.sleep(8.0)
self.send_magnet_command(0,"go")
time.sleep(2.0)
self.send_stop_belt_command()
time.sleep(1.0)
self.send_arm_motion_command("2")
time.sleep(4.0)
self.send_magnet_command(0,"stop")
self.send_move_command(28.0,100.0)
```



## 発生した問題



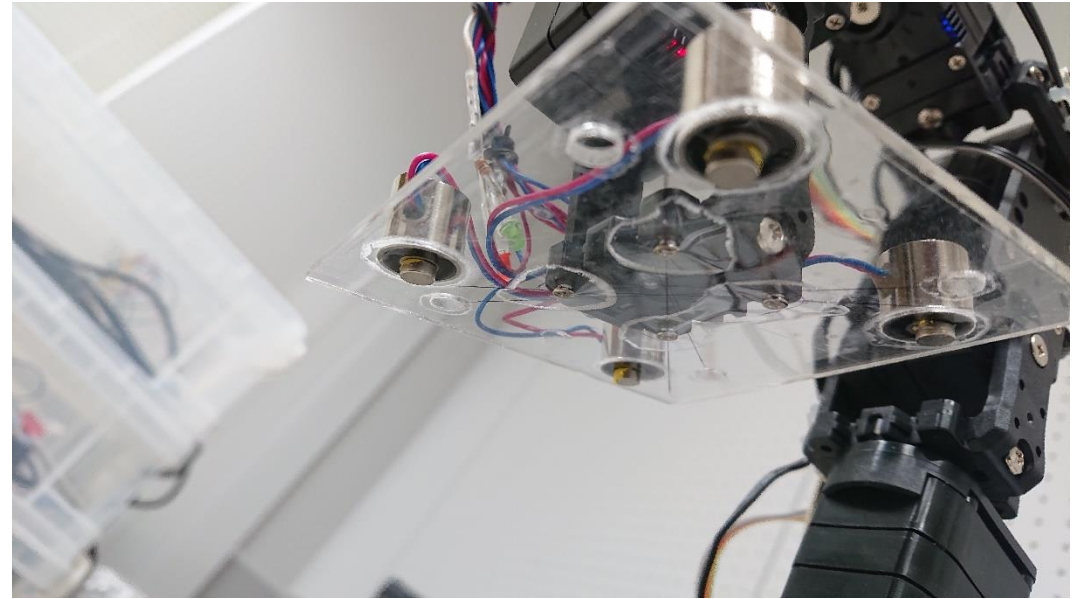
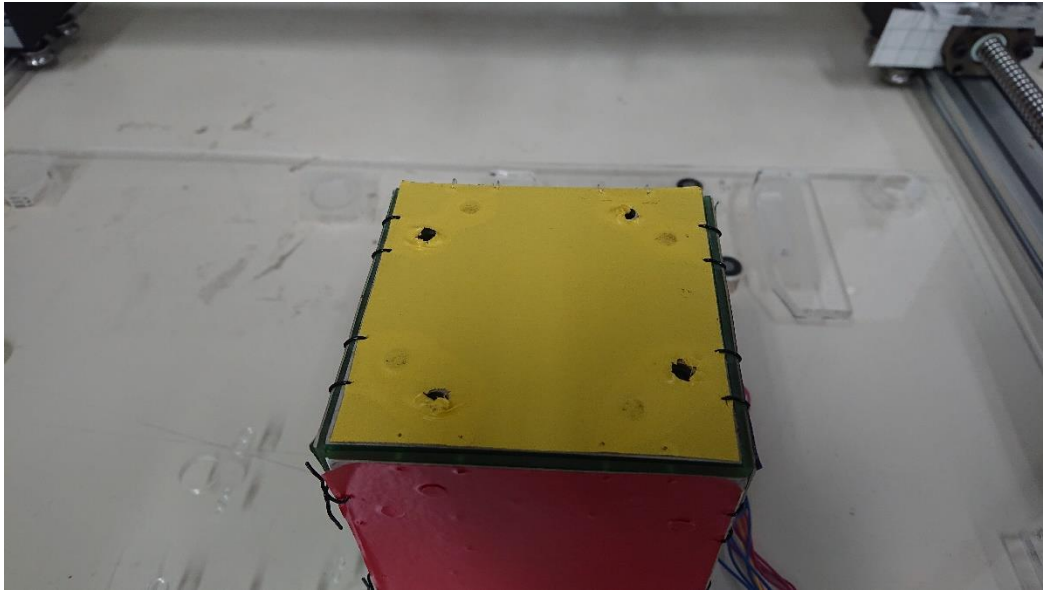
- ・ 事前の予想どおり、アームの電磁石とダイスが離れない  
- ちょっとずるしました。ガイドの貼り付け。



## 発生した問題



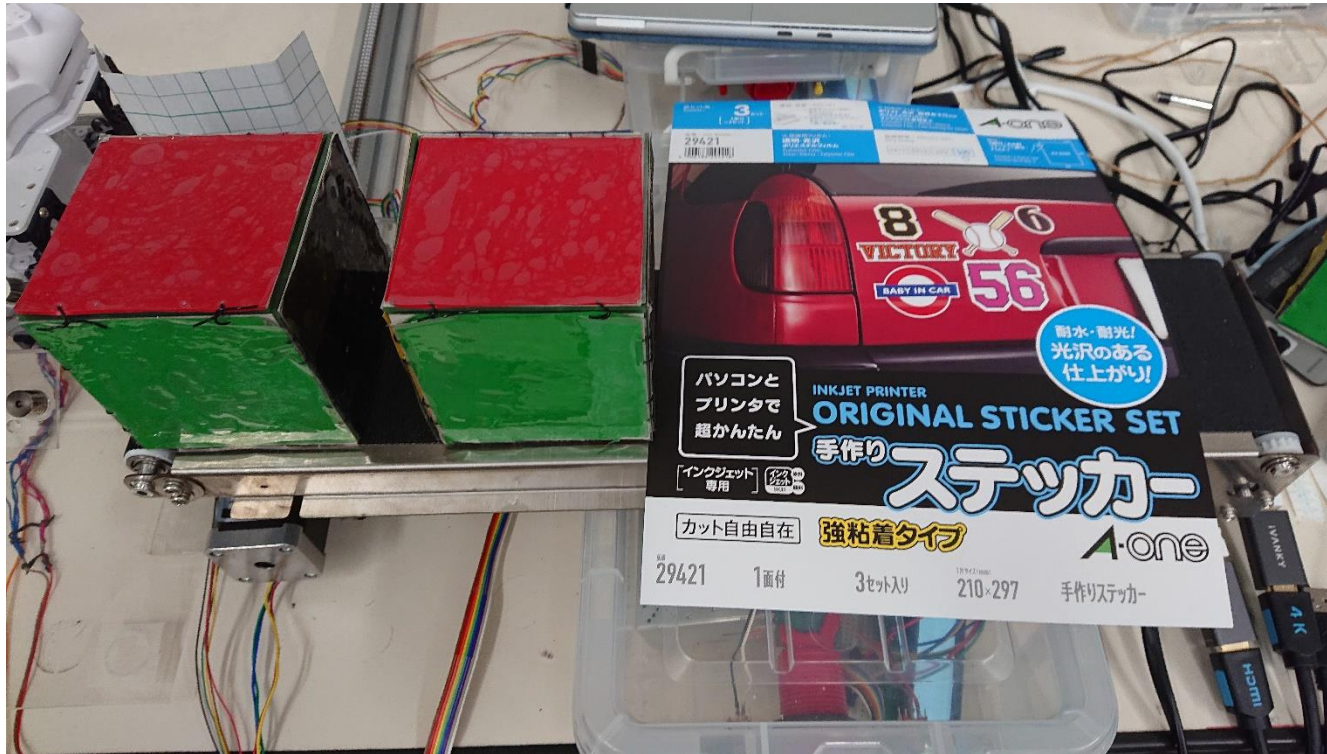
- ・ 電磁石が発熱して相手側のダイスの面の被服を溶かして、ネオジム磁石を剥がしてしまいました。



## 発生した問題



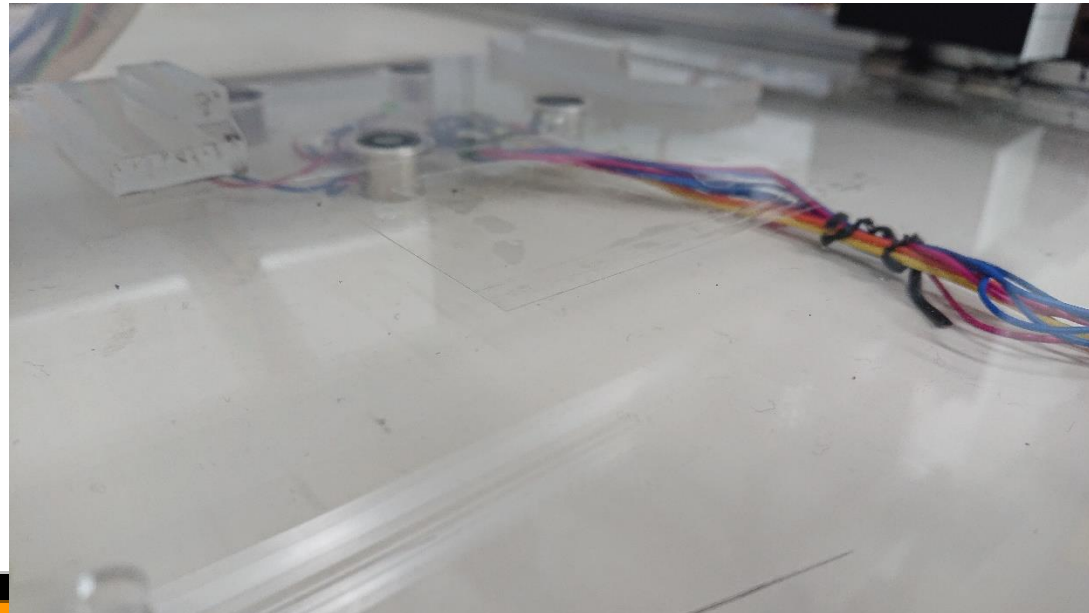
- ・ 熱対策… すべての面にステッカーの保護シールを貼りました



## 発生した問題



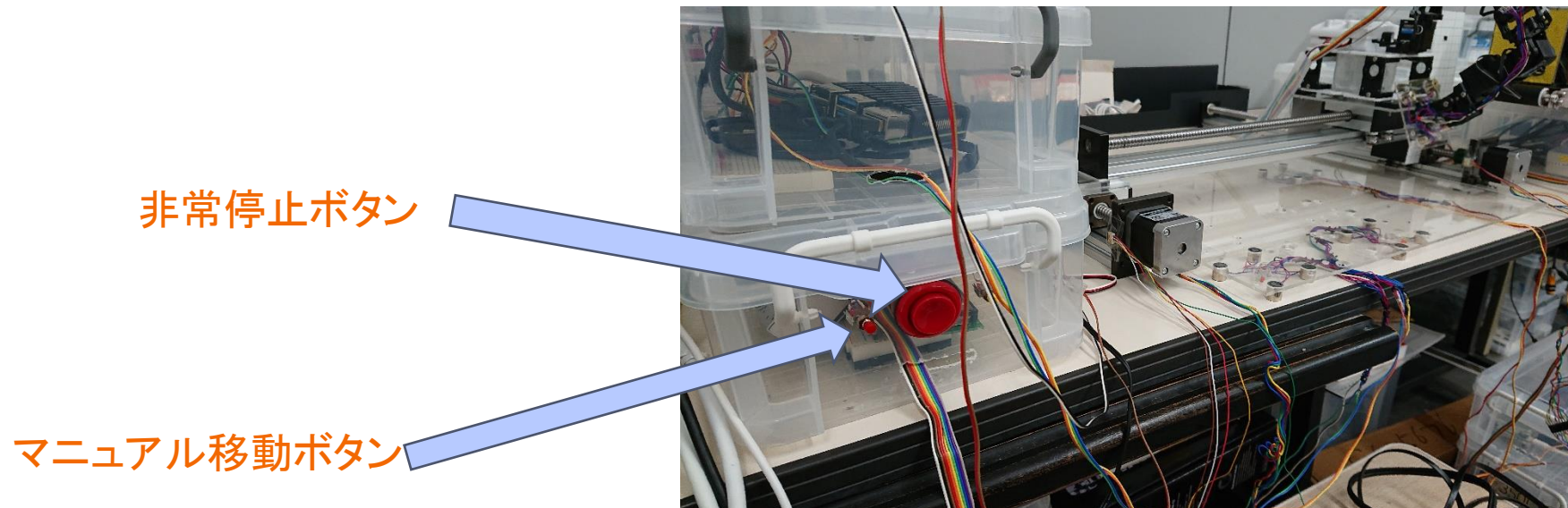
- ・ 回転場所の電磁石を張り付けているシールに、ダイスが引っ掛かって、意図した動作にならない。
  - ひっかかりがないように、別のシールを張り付け。



## 発生した問題



- 一部のlinear slide table が動かないことがある
  - >部品移動機構が壊れる
  - 非常停止ボタンを付加。ついでにマニュアル移動ボタンも付加。



## おわりに



- ・ アームロボットの動き(と電磁石の磁界の順方向/逆方向/停止の切り替え?)を使って、テスラディスク(部品)の把握、切り離し、垂直方向回転、水平方向回転、ができた。
- ・ (受信装置の方)あともう少しであってほしい
- ・ Virtual 空間の「物」から、それと同じものを組み立てるための受信装置の制御情報に変換できたら、「テレポーテーション」が、一応、実現できるはず。

# 謝辞



- What How For Kitakyushu



Project  
TESLA