

MUGBOT/Fabbotにセンサーを付ける

- ❑ センサーを付けると？
- ❑ センサー接続ボード
- ❑ 接続できるセンサー
- ❑ SensorBoardとScratchの接続
- ❑ センサーを使って
- ❑ 使いやすいセンサーを接続
- ❑ 使えそうなセンサー
- ❑ どのボードを使う？
- ❑ 使いたいセンサーを接続できるSensorBoardを作ろう
 - SensorBoard（改）の仕様－1
 - SensorBoard（改）の仕様－2
 - SensorBoard（改）の仕様－3
- ❑ SensorBoard（改）－3実装例
- ❑ 発話レスポンスの向上のために
- ❑ Arduinoピン割付け
- ❑ SensorBoard（改）－接続のまとめ
- ❑ 資料その他

* **MUGBOT**は東京都市大学 メディア情報学部社会メディア学科 小池情報デザイン研究室で作られたソーシャルロボットです。

* **Fabbot**はMUGBOTをベースに**ファブラボ**内で作られた簡単ロボットキットです。

* MAD(**Mugbot Action Designer**)は、MUGBOTを制御できるようにカスタマイズされたScratch1.4です。
MADは山梨英和大学の杉浦学先生の著作物です。

* 本書内の社名、製品名などは、一般に各社の商標または登録商標です。

* 本書内のイラストの多くは「**いらすとや**」の素材を使用しています。

センサーを付けると？

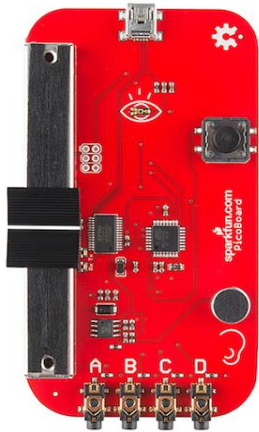
- センサー情報をプログラム内で読取れる
情報に対応して以降のプログラムの流れを変えられる

応用例：

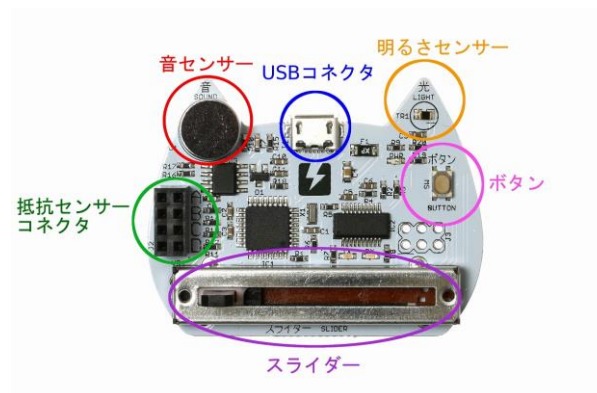
- 人が近寄ってきたら話し始める（自立案内）
- 操作者に選択させる（対話クイズ）

センサー接続ボード

- センサーを接続するためのScratch Sensor Board



スイッチサイエンス
PicoBoard



スイッチサイエンス
Necoboard2

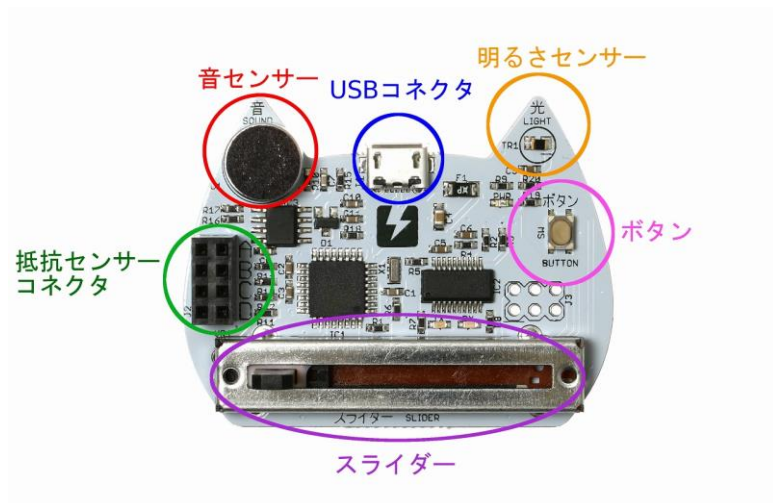


ちっちゃいものくらぶ
NanoBoard AG

「日本で買えるScratchセンサーボード」より

接続できるセンサー

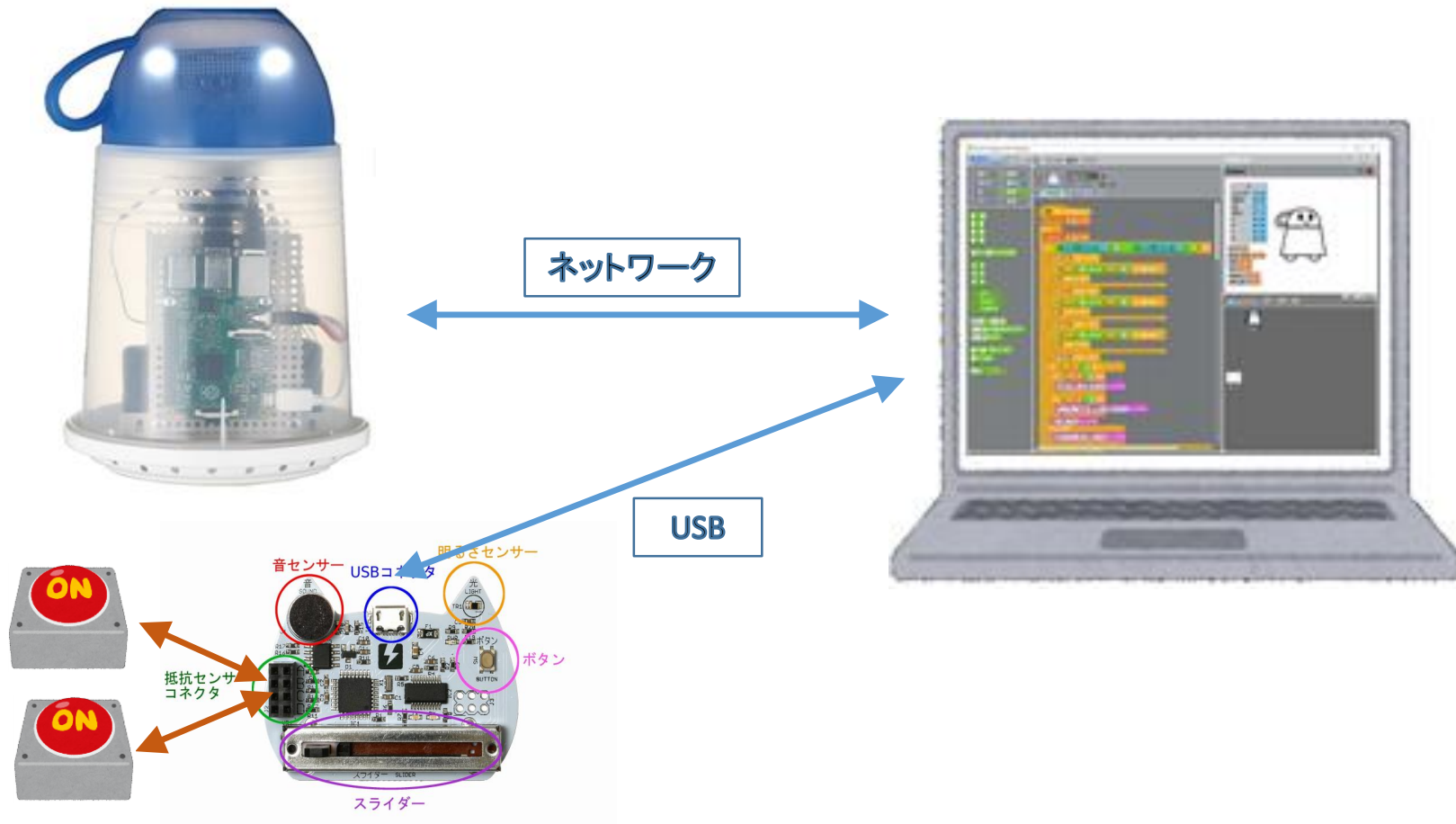
- いくつかのセンサーが接続できる
 - スライダーの位置（可変抵抗器）
 - 明るさ
 - 音の大きさ
 - ボタンの押下
 - 抵抗の接続（A、B、C、D）



入	
スライダー	93
明るさ	5
音	100
ボタン	false
A	93
B	92
C	3
D	4

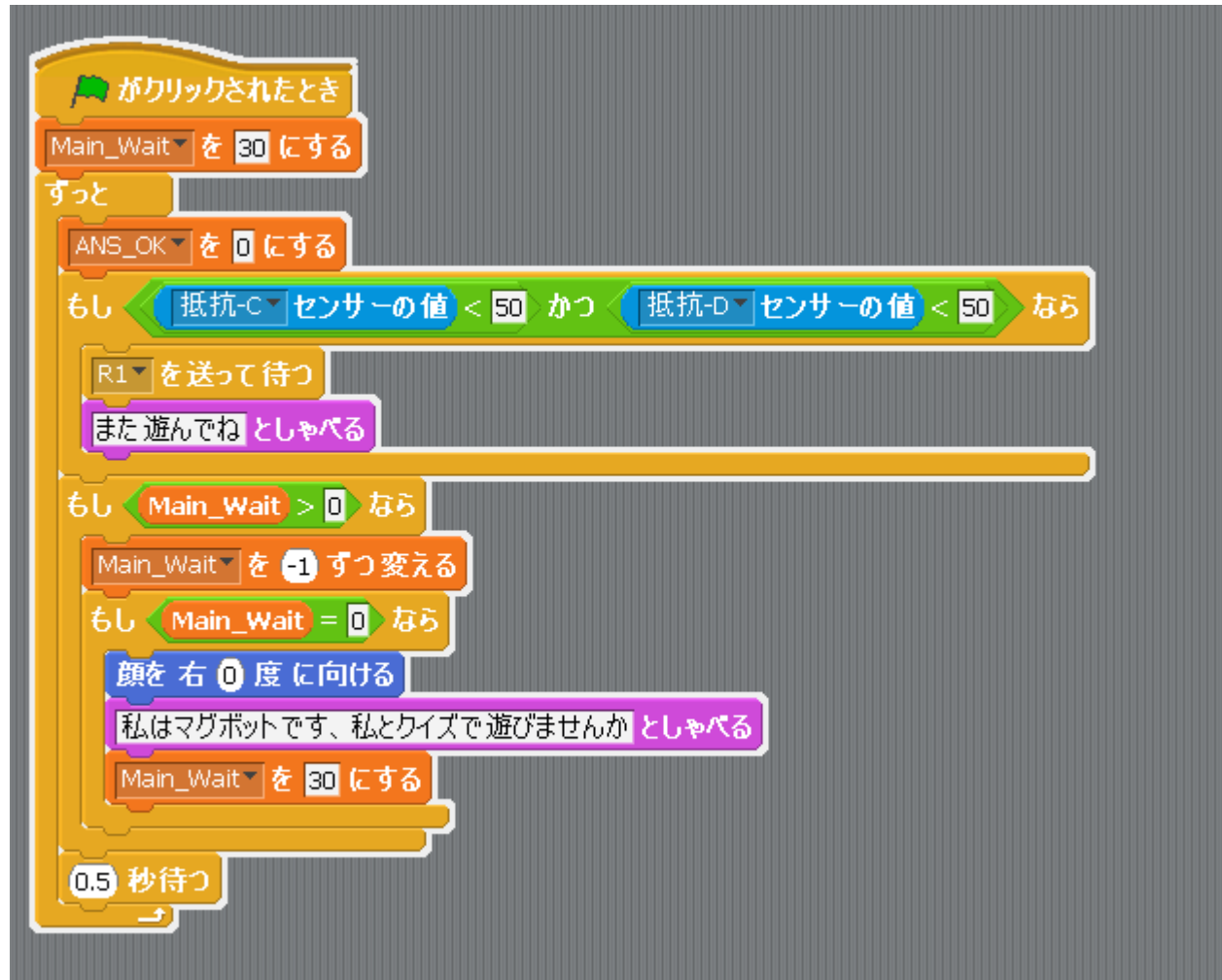
SensorBoardとScratch接続

- MUGBOT/Fabbotに接続した操作PCのUSBポートに接続



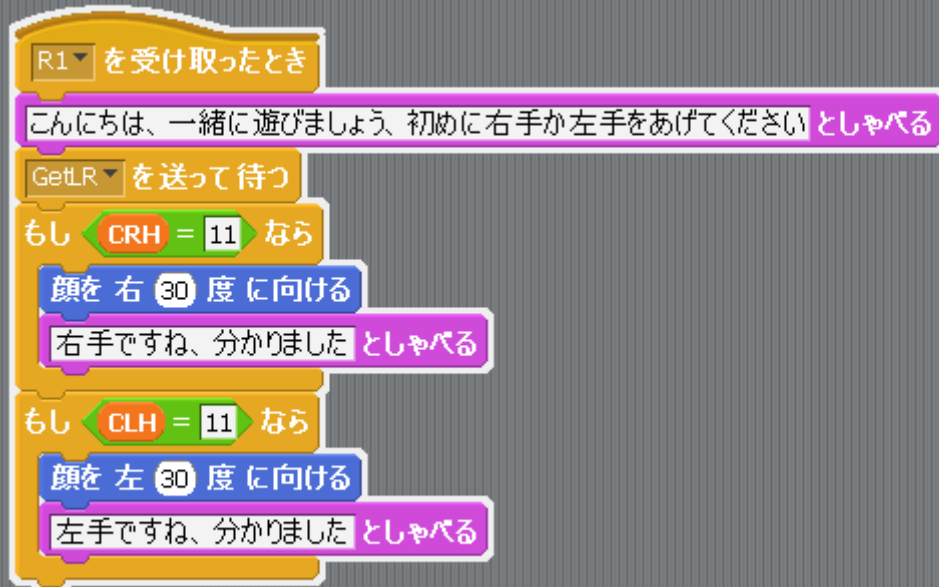
センサーを使って（１）

- 接続したセンサーの値を判断する



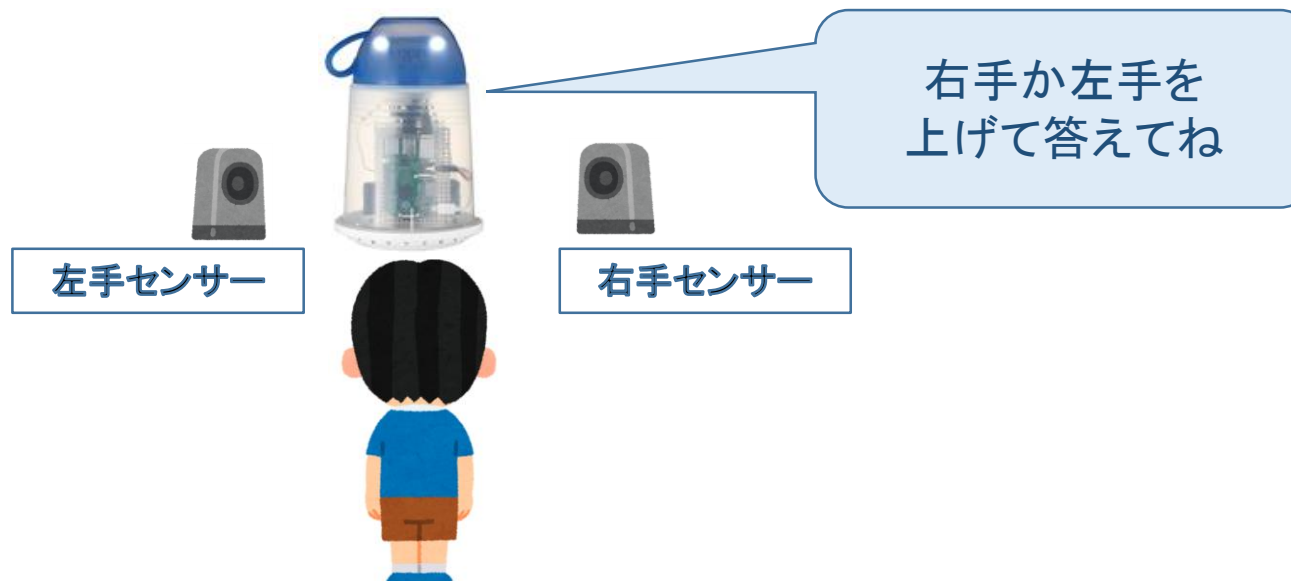
センサーを使って（２）

- 接続したセンサーの値を判断する



使いやすいセンサーを接続

- 押しボタンスイッチは見るからに「スイッチである」
接近センサーみたいな「スイッチっぽくない」のがいい



距離センサーを使って



- 左右の距離センサーの値を読み取り、右手、左手のどちらを上げたか判定する

使えそうなセンサー

・ 距離センサー／ジェスチャーセンサー



超音波センサー
HC-SR04

超音波の反射時間を計測して反射体までの距離を計測

- ・ **デジタル**
- ・ 2～400cm



赤外線センサー
GP2Y0A02YK

赤外線の反射時間を計測して反射体までの距離を計測

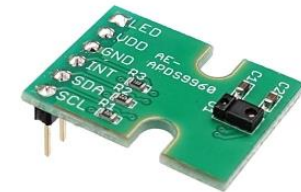
- ・ **アナログ**
- ・ 20～150cm



距離モジュール
GP2Y0E03

赤外線を使い三角法で距離を測定

- ・ デジタル+ **アナログ**
- ・ 4～50cm



ジェスチャーセンサー
NanoBoard AG

対象物との相対距離、周囲の明るさ、色、移動方向

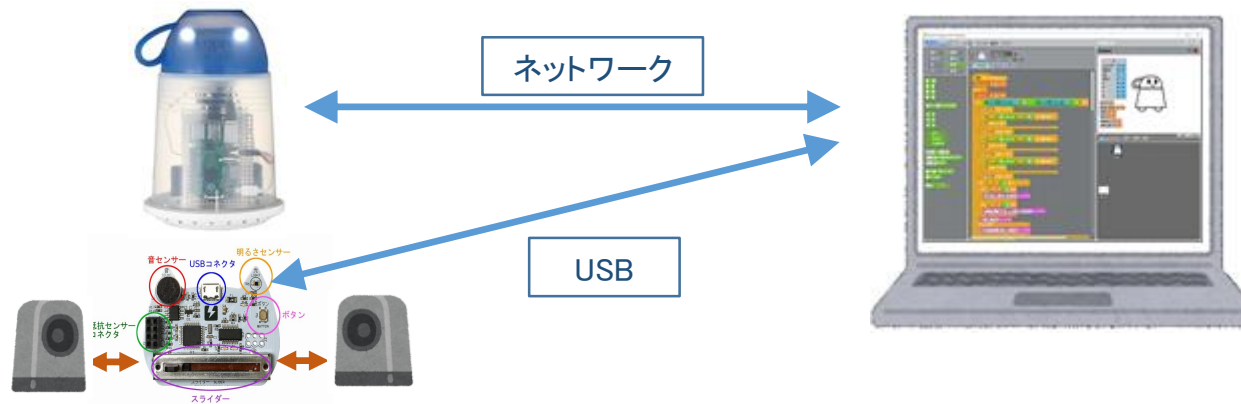
(ジェスチャー) を測定

- ・ **デジタル**(i2C)
- ・ 上下左右の移動方向

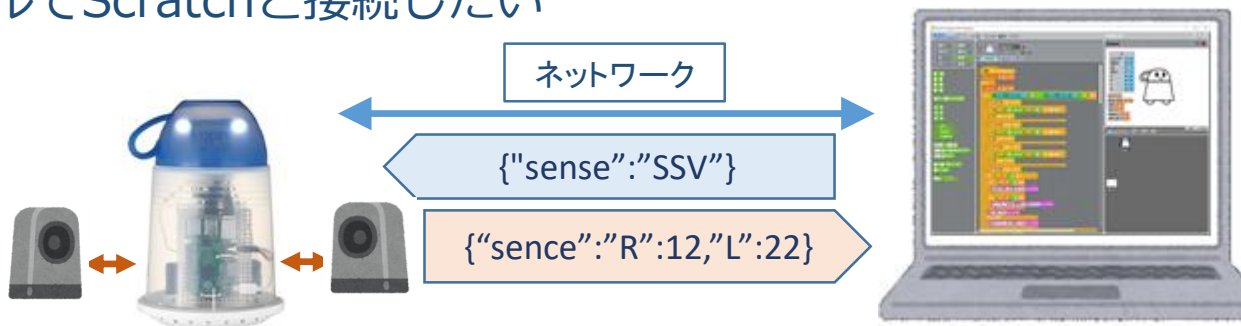


どのボードを使う？

SensorBoardには数種類あるが、どれもMUGBOTの操作PCに接続する



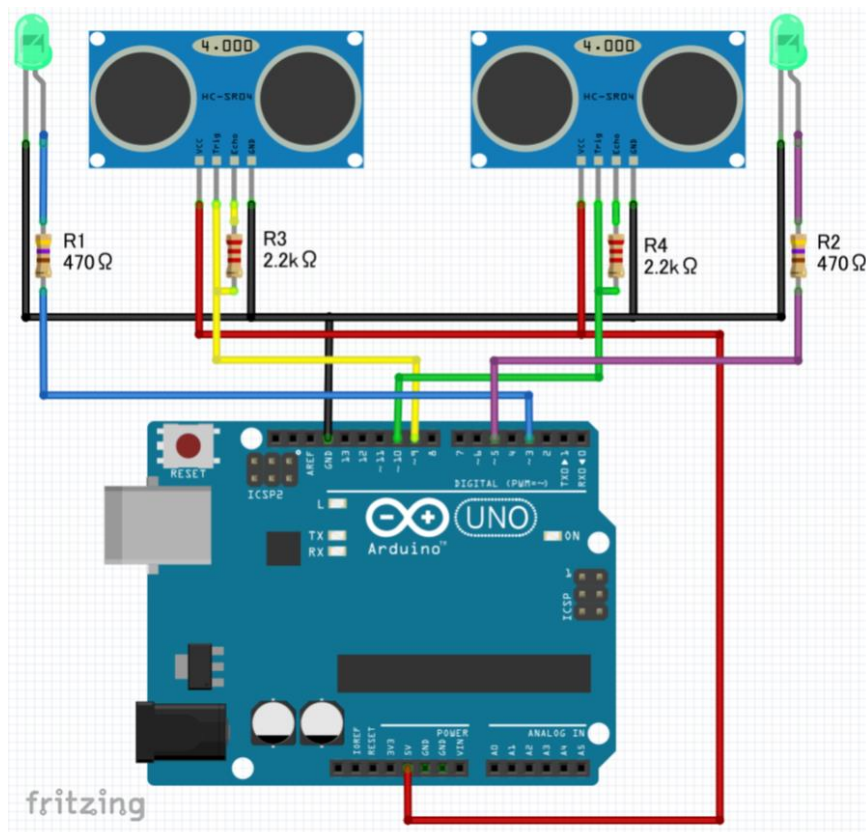
将来的にはMUGBOT本体に接続して、操作PCと接続したネットワークケーブルでScratchと接続したい



***MADの改造が必要になる**

SensorBoard (改) の仕様－1

- SecsorBoardと同仕様のボードを作る



操作PCがWindowsの場合：

- SensorBoardの通信はUSBポートを
操作PCのUSBポートに接続する

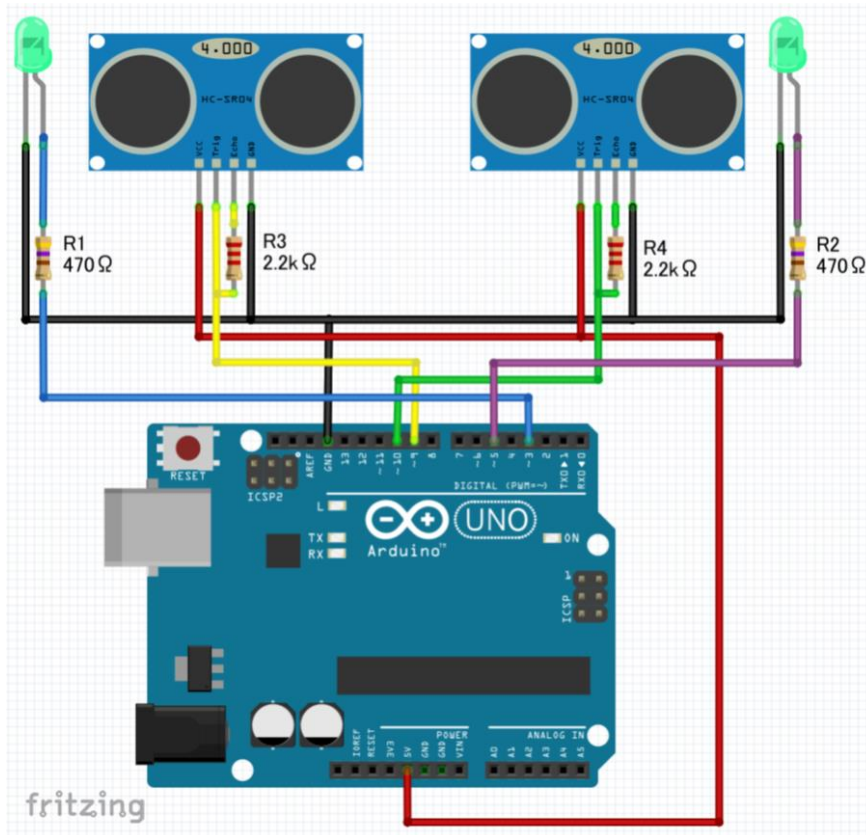
操作PCがmacOSの場合：

- SensorBoardの通信は別途USB-シリアルアダ
プタのTTL側をD0(RX)、D1(TX)に接続、
USB側を操作PCのUSBに接続

- 追加でArduinoが必要になる
- 操作PCがmacOSの場合、ArduinoのUSBポ
ートをSensorBoardとして認識しないため、
別途のUSB-シリアルアダプタが必要になる
- *nekoboard2やNanoBoard AGを使う方が
楽でいい

SensorBoard (改) の仕様－2

- MUGBOT/Fabbot本体の動作処理用Arduinoに組み込む



- センサー関連の追加だけを図示している、
動作のサーボ、LEDなどは現行のまま接続
動作のシリアルポートはD12(RX)、
D13(TX)をRasPiのTTLシリアルに接続

操作PCがWindowsの場合：

- SensorBoardの通信はUSBポートを
操作PCのUSBポートに接続する

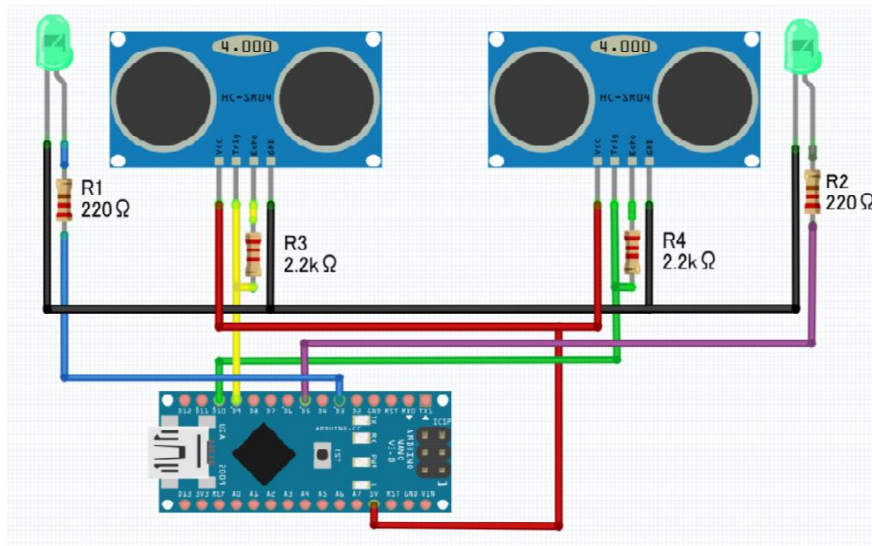
操作PCがmacOSの場合：

- SensorBoardの通信は別途USB-シリアルアダ
プタのTTL側をD0(RX)、D1(TX)に接続、
USB側を操作PCのUSBに接続

- 操作PCがmacOSの場合、ArduinoのUSBポー
トをSensorBoardとして認識しないため、
別途USB-シリアルアダプタが必要になる

SensorBoard (改) の仕様－3

- MUGBOT/Fabbot本体の動作処理用Arduino nanoに組込む



- センサー関連の追加だけを図示している、
動作用のサーボ、LEDなどは現行のまま接続
動作用のシリアルポートはD12(RX)、
D13(TX)をRasPiのTTLシリアルに接続
- SensorBoardの通信はUSB(B)を操作PCの
USB(A)に接続

- 現用のArduinoをArduino nanoに変更する
必要がある

SensorBoard (改) – 3 実装例

- MUGBOT/Fabbot本体の動作処理用Arduino nanoに組込む



発話レスポンスの向上のために

• RasPi上の発話処理(jsay)の改造

```
SETUP を受け取ったとき
色々と設定中です としゃべる
MUGBOT:私はマグボットです、私と一緒に遊びませんか としゃべる
CA:全部で としゃべる
CA1:一問 としゃべる
CA2:二問 としゃべる
CAALL:全部の質問に正解しました としゃべる
CAEND:に正解しました としゃべる
CANG:残念でした としゃべる
BY1:また遊んでね としゃべる
ANS:正解だと思ったら右手を上げてください、違っていると思ったら左手を上げてください としゃべる
QQ:分かりました としゃべる
QQOK:はい、正解です としゃべる
QQNG:残念でした としゃべる
R0:こんにちは、一緒に遊びましょう、初めに右手か左手をあげてください としゃべる
RR:右手ですね としゃべる
RL:左手ですね としゃべる
Q1:それでは問題です、私の名前はマグボットだと思いますか としゃべる
Q1NG:マグボットです としゃべる
Q2:次の問題です、今日は日曜日ですか としゃべる
Q2NG:今日は土曜日です としゃべる

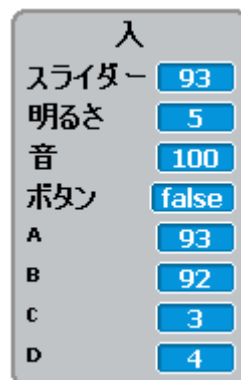
R1 を受け取ったとき
:R0 としゃべる
GetLR を送って待つ
もし CRH = 11 なら
顔を右 30 度に向ける
:RR:QQ としゃべる
もし CLH = 11 なら
顔を左 30 度に向ける
:RL:QQ としゃべる
```

- Scratch上での発話は「〇〇としゃべる」でRasPiに「talkコマンド」として送られ、逐次jtalkを使って.wavファイルを生成し、それを再生することで処理される
- 対話型のプログラムを使った場合、この変換時間のためにセンサーからの信号で対応する音声の流れ始めるまでの遅れが気になる、特に処理速度の遅いRasPiの場合の遅れは大きい
- このために発話処理スクリプト(jsay)を拡張する試みを行った
 - 発話文字列の仕様拡張で実装する
発話データの定義（登録）：
データ名:発話データ
発話データの再生：
:データ名1[:データ名2]...
定義例)
CA:全部で
CA1:一問
CA2:二問
CAEND:に正解しました
再生例)
:CA:CA1:CAEND ⇒「全部で 一問に 正解しました」
*発話データ内に「:」が含まれない場合はそのまま発話

Arduinoピン割付け

pin	用途
D0	RX
D1	TX
D2	Servo（顔上下）
D3	LED（左手センサー）
D4	Servo（顔左右）
D5	LED（右手センサー）
D6	LED（左目）
D7	SW（ボタン・センサー）
D8	LED（口）
D9	左手センサー信号（D）
D10	右手センサー信号（C）
D11	LED（左目）
D12	Software Serial（RX）
D13	Software Serial（TX）

pin	用途
A0	スライダー
A1	
A2	抵抗(B)
A3	抵抗(A)
A4	明るさセンサー
A5	音量センサー



入

スライダー	93
明るさ	5
音	100
ボタン	false
A	93
B	92
C	3
D	4

SensorBoard (改) – 接続のまとめ

- 単独SensorBoardの場合 ()内はピン番号

Board	Windows	macOS
UNO	USB	ExUSB(0,1)
Leonardo	USB	ExUSB(0,1)
Nano	USB	USB

- MUGBOT/Fabbot本体の動作処理用Arduinoに組み込む場合 ()内はピン番号

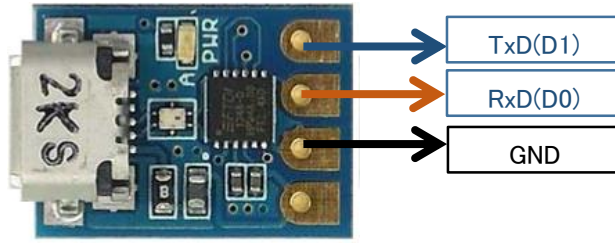
Board	Windows		macOS	
	BC	SB	BC	SB
UNO	SS(12,13)	USB	SS(12,13)	ExUSB(0,1)
Leonardo	SS(12,13)	USB	USB	ExUSB(0,1)
Nano	SS(12,13)	USB	SS(12,13)	USB

*BC:BOT Control(MUGBOT制御通信)、SB(Sensor Board通信)

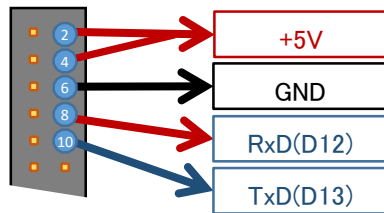
SS:Software Serial(RX,TX)、ExUSB:External USB Adapter(RX,TX)

資料その他

- [USB-シリアルアダプタ](#)の接続



- [Raspberry pi](#)のTTLシリアル



+5Vはここから供給しても良い（Micro USB側は無効になる）

設定などは下記を参照

[Rubyを使ってRaspberry Pi3でシリアル通信\(UART通信\)をする](#)

Raspberry piの信号レベルは3.3V、Arduinoは5Vであることに注意（抵抗分圧などでレベル変換が必要）

Device名は"/dev/serial0"(現行のUSBポートは"/dev/ttyACM0")