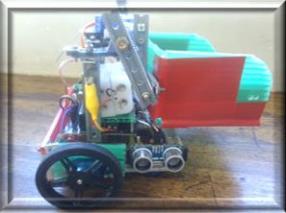
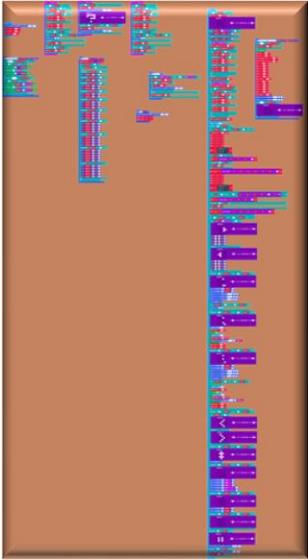
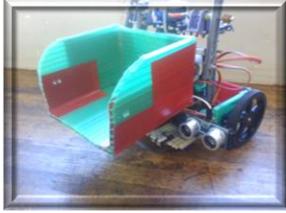




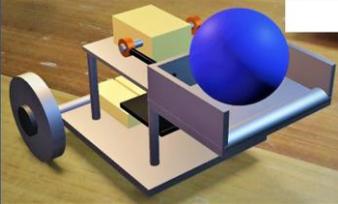
学校名	谷田部東中学校			この作品はクリエイティブ・コモンズ表示4.0国際ライセンスの下に提供されています。
(ふりがな)	しょうろんぼうしぐま			
チーム名	Shorompo + Σ			
ロボコンルール (名称とURL)	創造アイデアロボットコンテスト 制御・計測 部門 http://ajgika.ne.jp/~robo/	都道府県名	茨城県	
製作期間	2021年 4月頃から 2021年 11月頃まで	製作時間	0÷0 時間	
ロボットに関する写真と図 必ず、ロボットの概要や機構等の特徴がわかる写真や図等を1~4枚で掲載する。 写真や図に記号等を書き込み、下の枠「ロボットのアイデア概要」で解説する。	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  <p>▲アームを動かす部分</p> </div> <div style="text-align: center;">  <p>▲プログラム</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;">  <p>←左</p> </div> <div style="text-align: center;"> <p>←右</p> </div> </div>			
ロボットのアイデア概要【報告書要約】 どのような動きを実現するために、具体的にどのような素材や機構を用いて実現したのか、枠いっぱい解説を書き込むこと。	<ul style="list-style-type: none"> • 収穫ゾーンに行くために回転サーボモーターをタイヤに使った。 • ボールを収穫するためにサーボモーターを使ってアームを作った。 • スタートゾーンに戻ってくるために開店サーボモーターをタイヤに使った。 ○ 収穫物置場にボールを入れるためにサーボモーターでアームを作った。 • これらを制御するためにMicrobit Kitronik Robotics Boardをつかった。 			
参考資料 製作上参考にした資料や、参考にした先輩のロボット等の情報についてできるだけ詳しく解説する。	<ul style="list-style-type: none"> • MicrobitKitronikRoboticsBoardの使い方 5641-microbit-robotics-board-datasheet (kitronik.co.uk) 			

報告書の2枚目以降にさらに詳しく自由フォーマットで記入する。この用紙を入れて6枚以内で報告書を作成すること。

Shorompo + Σの軌跡

3Dモデル

ロボットの作成に当たって、完成のイメージとなる3Dモデルを作った。

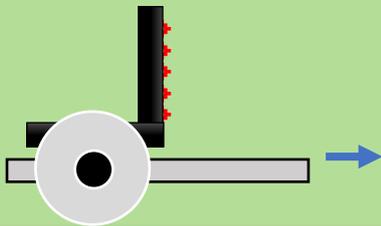


最初期の予定。左右のタイヤで移動し、上についたサーボモーターでアームを動かす形。

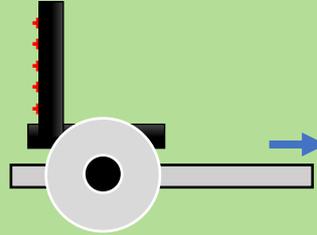
車体

① ロボティクスボードの配置：ロボティクスボード[Micro:bit]をどこに置くかが問題だった。だから、度々置く場所を変えていた。

(I) 前向き



(II) 後ろ向き



一応操作性も考えて、後ろ向きにつけることにした。

② タイヤ：グリップ力と速度、そしてバランスを保てるタイヤを選んだ。

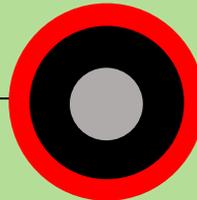
タイヤ。うまく写っている写真がなかったため絵で代用

選んだもの→



ほかの候補だったもの→

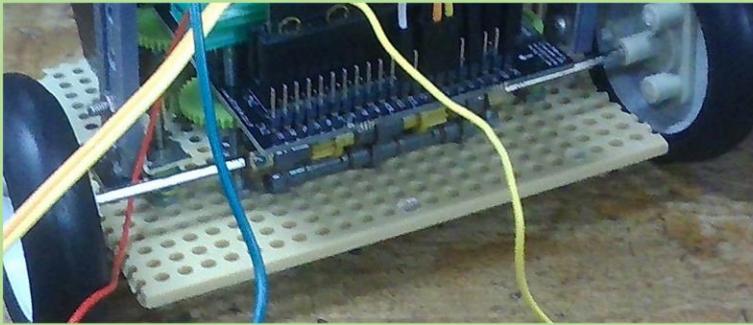
風船をタイヤに巻く



車体

③モーター：

最初はタイヤのみDCモーターで動かして、
アームはサーボモーターで動かす予定だった。



DCモーターを入れたツインギアボックスでタイヤのシャフトを回す算段。

しかし、左右でスピードがずれてしまうという問題を抱えていた。

そして、モーターに負荷がかかると理由はわからないが、超音波センサーに間違った電流が流れ、まともに走れなくなってしまうこともあり、後に360°回転サーボに変更した。



←↓サーボモーター。



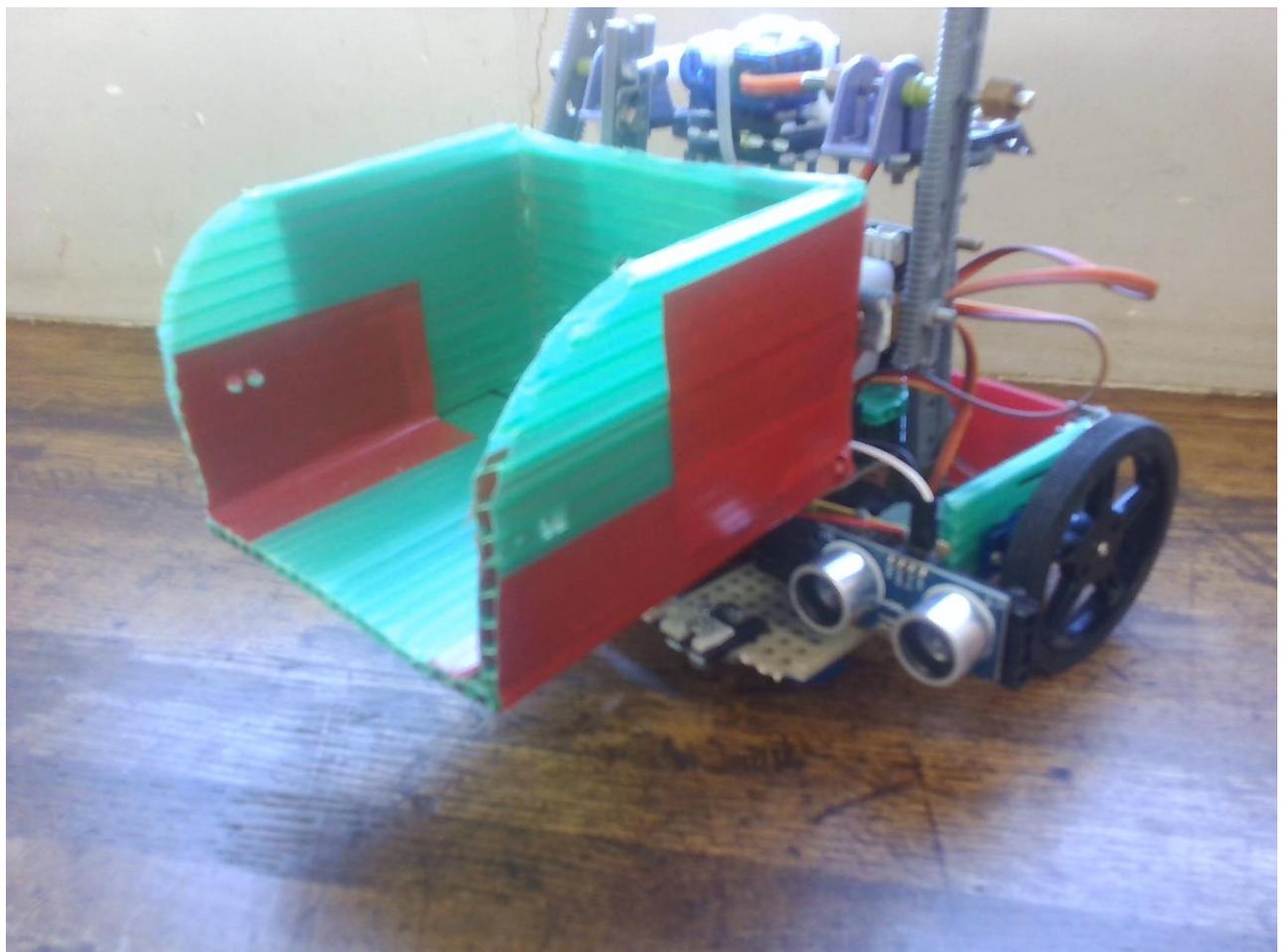
④センサー：

超音波センサーを使った。



←超音波センサー。大体200cmぐらいまで測れる

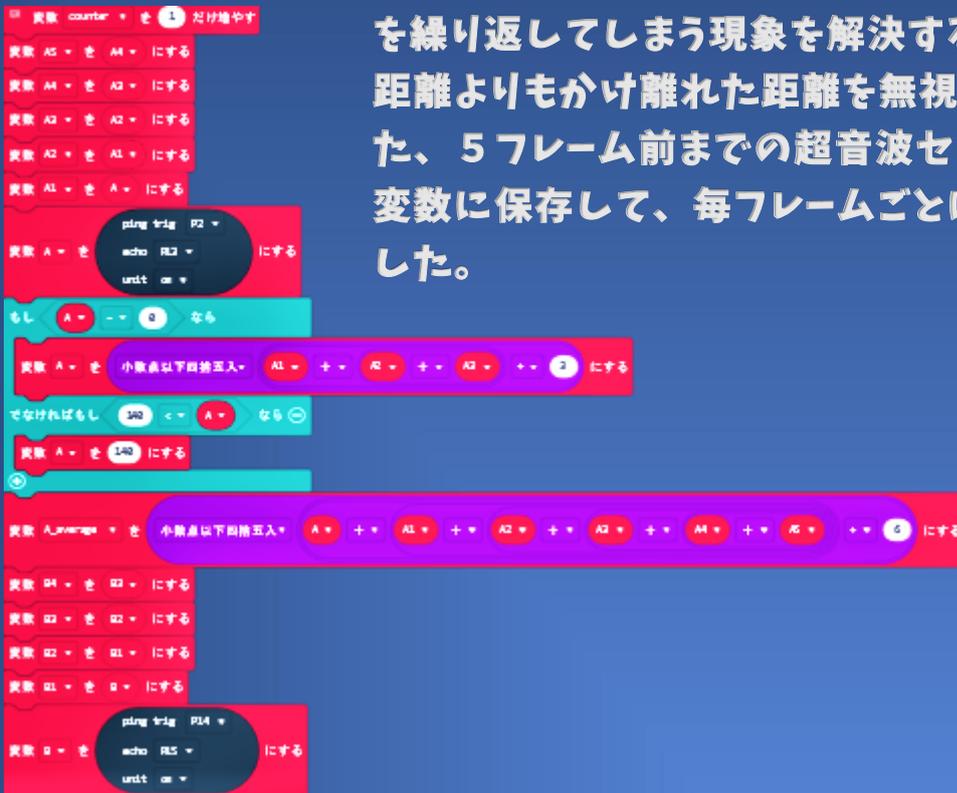
プログラムは拡張機能の「sonar」を使ってプログラミングする。



↑完成形

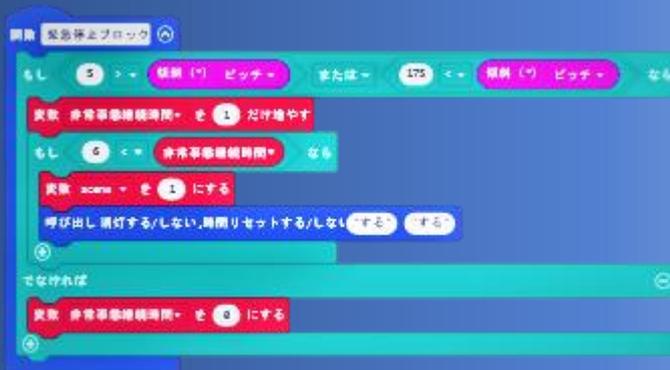
④プログラミングの解説

- ・超音波センサー問題：超音波センサーの前にある物体があまりにも近いときに測定した距離が無限に大きくなったり0になったりを繰り返してしまう現象を解決するため、測定すべき距離よりもかけ離れた距離を無視するようにした。また、5フレーム前までの超音波センサーの測定結果を変数に保存して、毎フレームごとに平均を出すようにした。



```
関数 counter を だけ増やす
関数 A5 を A4 にする
関数 A4 を A3 にする
関数 A3 を A2 にする
関数 A2 を A1 にする
関数 A1 を A にする
ping trig P2 を
echo R2 を にする
関数 A を 0 にする
もし A < 5 なら
関数 A を 小整数以下四捨五入 A1 + + A2 + + A3 + + A4 + + A5 + + 2 にする
でなければもし 340 < A なら
関数 A を 140 にする
関数 Average を 小整数以下四捨五入 A + + A1 + + A2 + + A3 + + A4 + + A5 + + 5 にする
関数 R4 を R3 にする
関数 R3 を R2 にする
関数 R2 を R1 にする
関数 R1 を R にする
ping trig P14 を
echo R5 を にする
関数 R を 0 にする
```

- ・緊急停止：もしロボットがひっくり返ってしまったときに、アームをそのまま動かしてしまうと壊れる可能性があるので、傾きを計測してモーターを止めるプログラムを作った。



```
緊急停止ブロック
もし 5 > 傾斜 (°) ピッチ を または 175 < 傾斜 (°) ピッチ なら
関数 非常事態継続時間 を だけ増やす
もし 5 < 非常事態継続時間 なら
関数 scene を 1 にする
呼び出し 実行する/しない、時間リセットする/しない 実行する
でなければ
関数 非常事態継続時間 を 0 にする
```