

```

065 while Infinite Loop
066   while Infinite Loop
067     LED-L = on
068     LED-R = on
069     CN03 > 40%
070     if
071       Left = -50% Right = -50%
072       Wait = 1.0 sec
073       A = 0
074       Left = -50% Right = 50%
075       Wait = 1.0 sec
076       Left = 50% Right = 50%
077       Wait = 1.0 sec
078       Left = -50% Right = 50%
079       Wait = 1.0 sec
080       Left = -50% Right = -50%
081       Wait = 2.0 sec
082       A = A + 235711319
083     else
084       Left = 50% Right = -50%
085       Wait = 1.0 sec
086       Left = 50% Right = 50%
087       Wait = 1.0 sec
088       Left = 50% Right = -50%
089       Wait = 1.0 sec
090       Left = -50% Right = -50%
091       Wait = 2.0 sec
092       A = 0
093     end if
094   else
095     CN05 < 20%
096     Left = 10% Right = 30%
097     Wait = 0.2 sec
098     Left = 0% Right = 0%
099     Wait = 0.5 sec
100     CN01 < 9%
101     for [x 5]
102       (LED3)RED = on
103       Wait = 0.5 sec
104       (LED3)RED = off
105       Wait = 0.5 sec
106     end for
107     else
108     for [x 5]
109       (LED1)GREEN = on
110       Wait = 0.5 sec
111       (LED1)GREEN = off
112       Wait = 0.5 sec
113     end for
114     Left = 60% Right = 55%
115     Wait = 0.4 sec
116   else
117     CN06 < 9%
118     Left = 30% Right = 10%
119     Wait = 0.2 sec
120     Left = 0% Right = 0%
121     Wait = 0.5 sec
122     CN02 < 9%
123     for [x 5]
124       (LED3)RED = on
125       Wait = 0.5 sec
126       (LED3)RED = off
127       Wait = 0.5 sec
128     end for
129     else
130     for [x 5]
131       (LED1)GREEN = on
132       Wait = 0.5 sec
133       (LED1)GREEN = off
134       Wait = 0.5 sec
135     end for
136     Left = 55% Right = 60%
137     Wait = 0.4 sec
138   else
139     Left = 31% Right = 25%
140     (LED3)RED = off
141     (LED2)AMBER = on
142     (LED1)GREEN = off
143   end if
144 end if
145 end if
146 end if
147 end if
148 end while

```

069~  
blackセンサーが壁を検知したら方向転換

72~092  
A=0 つまり曲がる回数が奇数回の時は左に、それ以外の時は右に曲がるようにして同じ場所を探索しないようにした。こうすることで、誤発見も防げる。  
また、後退を2秒するのは機体を壁と平行にして、少し斜めになっているため安定して直進ができるため。  
(一番下に詳しい解説)

094~138  
CN5が20を下回る、つまり被災者を検知したときは後ろにあるCN1のセンサーにも検知させるため少し前に進ませる。  
もしCN1のセンサーが9を

下回る、つまり黒だった場合は赤ランプを上回る、つまり緑だった場合は緑ランプを

点滅させて被災者の識別をする。同様に、CN6でも同じことをする。CN5と6をどちらも使うことで検知の幅が広くなり、より多くの被災者を発見、正確に識別できるようになった。  
また、検知し終わった後の誤発見を防ぐために少し前進させるようにした。

CN140~  
上記のどれにも当てはまらない場合は前進をする。

## ライトレース

```

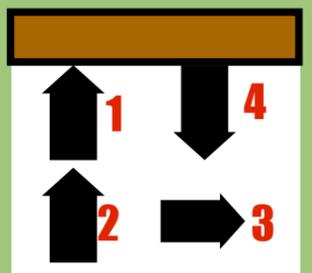
001 LED-L = on
002 LED-R = on
003 while CN01 < 60%
004   if CN04 > 17%
005     Left = -50% Right = -50%
006     Wait = 0.7 sec
007     Left = -50% Right = 50%
008     Wait = 1.0 sec
009     while CN06 > 20%
010       Left = 50% Right = 20%
011     end while
012     Left = 50% Right = 50%
013     Wait = 0.3 sec
014     Left = -50% Right = 50%
015     Wait = 0.2 sec
016   else
017     if CN05 < 3%
018       if CN06 < 3%
019         Left = 53% Right = 53%
020         Wait = 0.1 sec
021       else
022         Left = -50% Right = 70%
023         Wait = 0.1 sec
024       end if
025     else
026       Left = 70% Right = -50%
027       Wait = 0.1 sec
028     end if
029   end if
030 end while
031 Left = 50% Right = 50%
032 Wait = 0.5 sec
033 end while

```

003  
ホワイトラインセンサーがアルミホイールを読み取ったとき、レスキューのプログラムに移行。

004~015  
障害物をセンサーで検知したら後ろに下がり、避ける

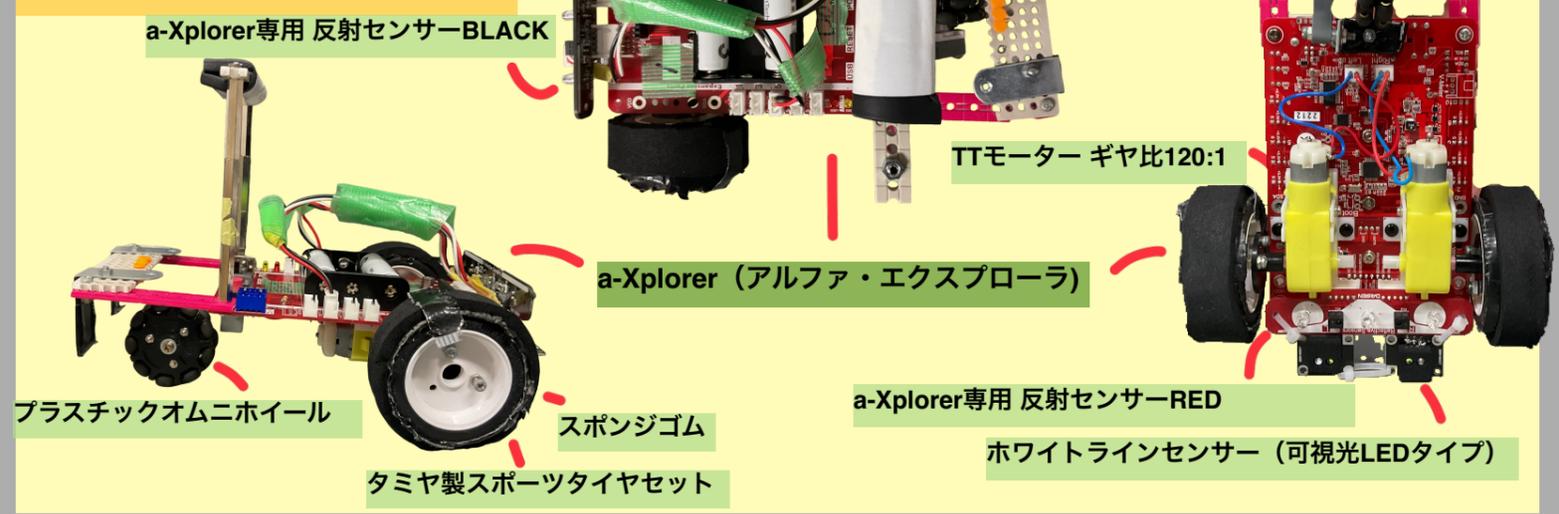
017~  
前二つのセンサーを使ってライトレース



## レスキュー 方向転換の仕組み

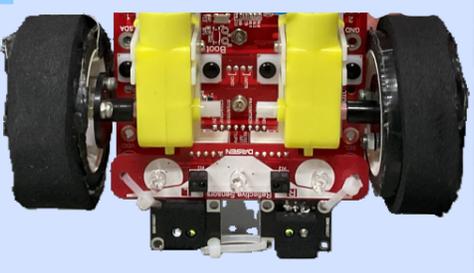
- 1 センサーで壁を検知する。
- 2 後ろに下がる。(3でロボットが曲がる時、壁にぶつからないようにするため)
- 3 今まで曲がった回数が、奇数だと左、偶数だと右に曲がって、そのあと少し前進する。(上図は偶数の時)
- 4 壁を背に向けるよう曲がって、そのあと後退し、ロボットを壁に垂直にする。(これにより、安定して直進することができる)

## 基本情報



## ロボット

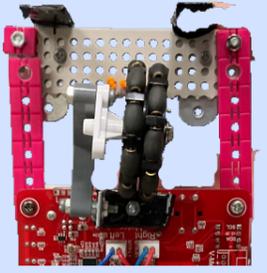
被災者の黒と緑を区別するとき、明るさによってセンサーが読み取る数値が変わらないように、明かりをつけながら走行する。



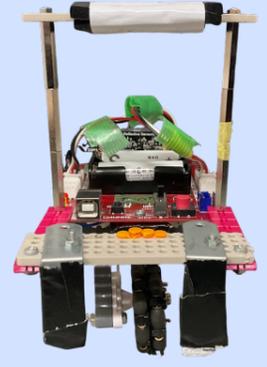
タイヤにスポンジを巻きつけ、摩擦力を増やしたり、タイヤを大きくして坂やバンプを上りやすくした。

前方のホワイトラインセンサーはライトレース用に床と被災者を識別するために、後方のREDセンサーは緑と黒の被災者を区別するために使う。緑は黒より値が高くなる。

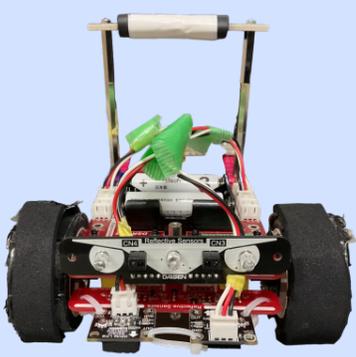
オムニホイールの高さをいつでも調整できるようにした。



後輪をオムニホイールにすることで、バンプも通常の速さで乗り越えられるようになった。



後ろにL字金具をつけることで、後ろに壁がある時、ロボットが壁に垂直になるようにした。



前の反射センサーで障害物や、レスキューエリア内の壁を検知する。



RCL-008