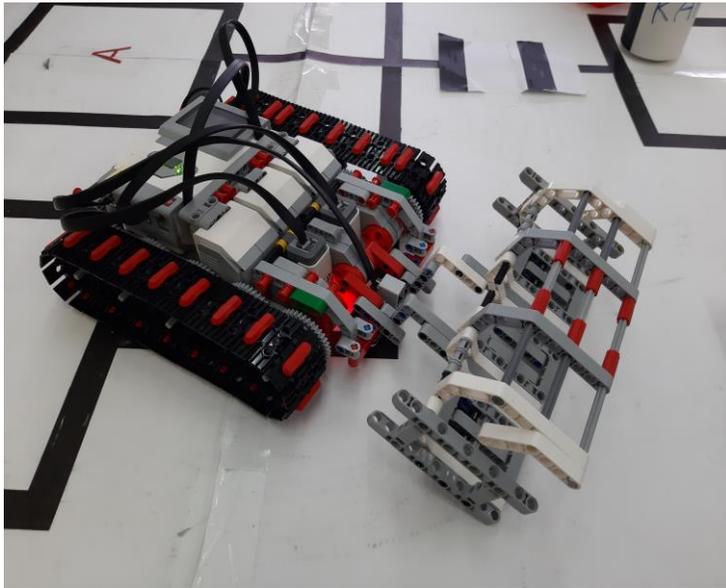


RoboCupJunior Rescue Line



Инженерный журнал

Команда: Котофалк

Состав команды:

Покумин Никита,

Демидов Сергей,

Сурков Яков

МАУ ДО "ПЦ ДОД "ШТР"

ОГЛАВЛЕНИЕ

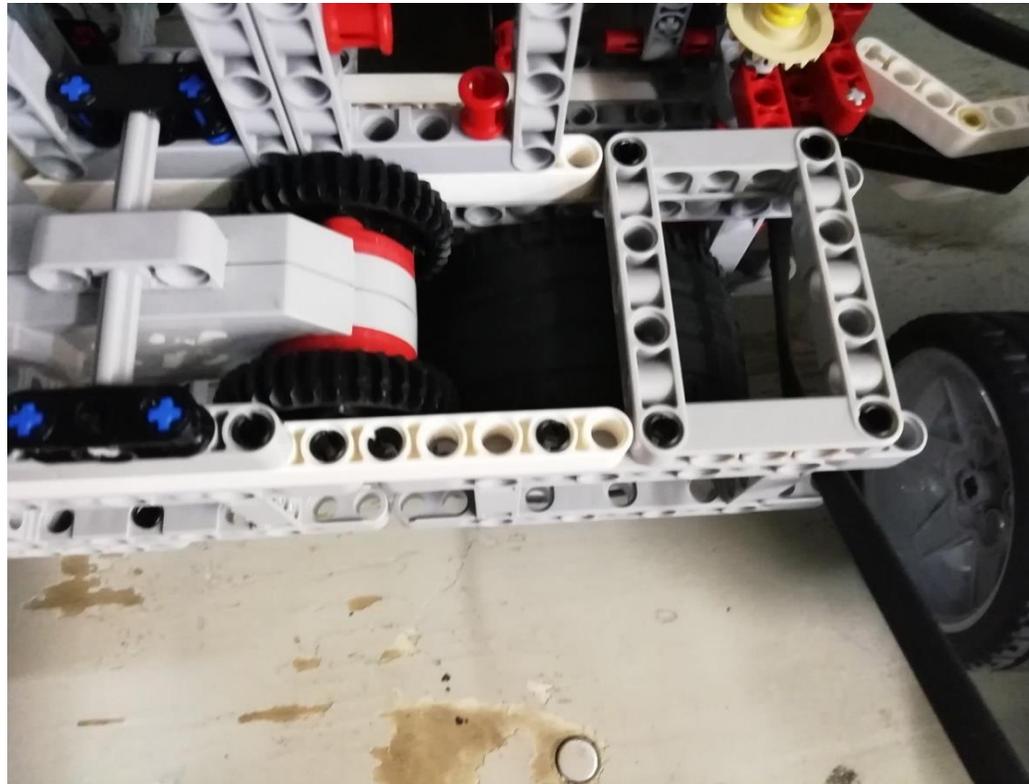
1. Конструкция робота.....	3 стр.
2. Блок- схема программы	6 стр.
3. Код программы.....	9 стр.

КОНСТРУКЦИЯ РОБОТА

1. Разработать конструкцию робота для прохождения всех препятствий (балки, горки и прочее)

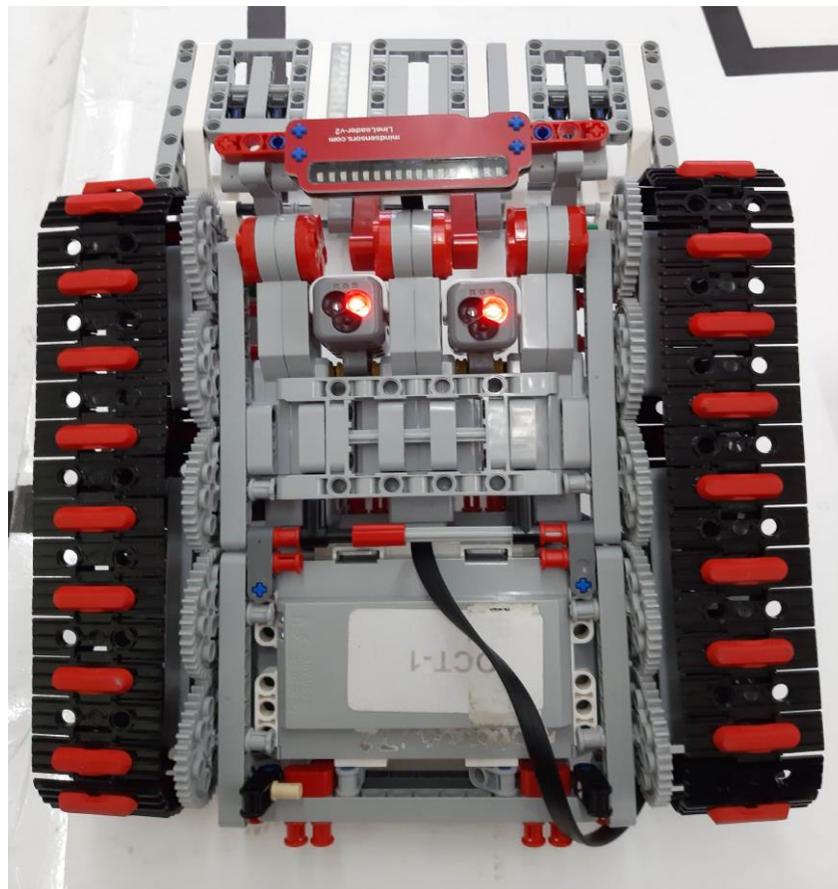
Для прохождения дверных проемов и для въезда в зону эвакуации было решено собрать максимально компактного робота.

Была разработана конструкция, которая при помощи силы трения передает вращение с сервопривода на ведущее колесо.



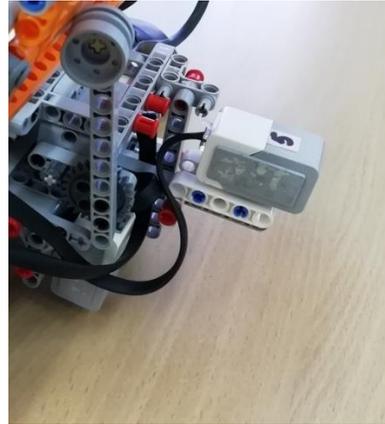
Данная конструкция помогла нам выигрывать в компактности робота, но проигрывала в функциональности: робот не мог преодолевать препятствия и при поворотах прокручивал колеса и не выполнял заданное условие.

Было решено собрать гусеничную конструкцию.



Данная конструкция полностью удовлетворяла нашим требованиям: робот четко выполнял заданные условия и с легкостью преодолевал препятствия.

Первоначально датчики цвета мы жестко зафиксировали к конструкции робота.



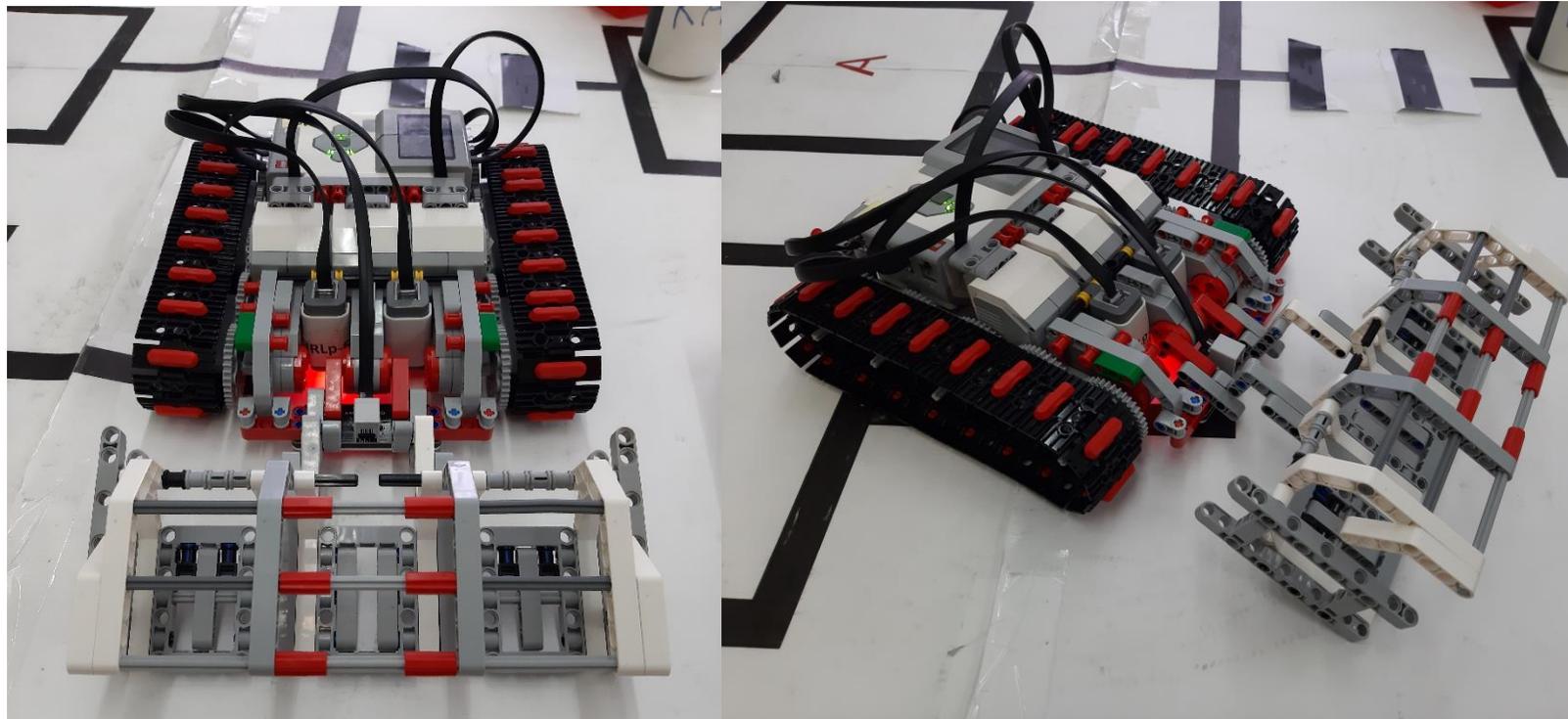
Но при прохождении препятствий, робот считывал цвет не верно и сбивался, из-за чего уезжал с заданного маршрута. Мы приняли решение сделать датчики независимыми от конструкции робота и адаптированными к любым препятствиям. И разместили их так, чтобы черная линия была между ними.



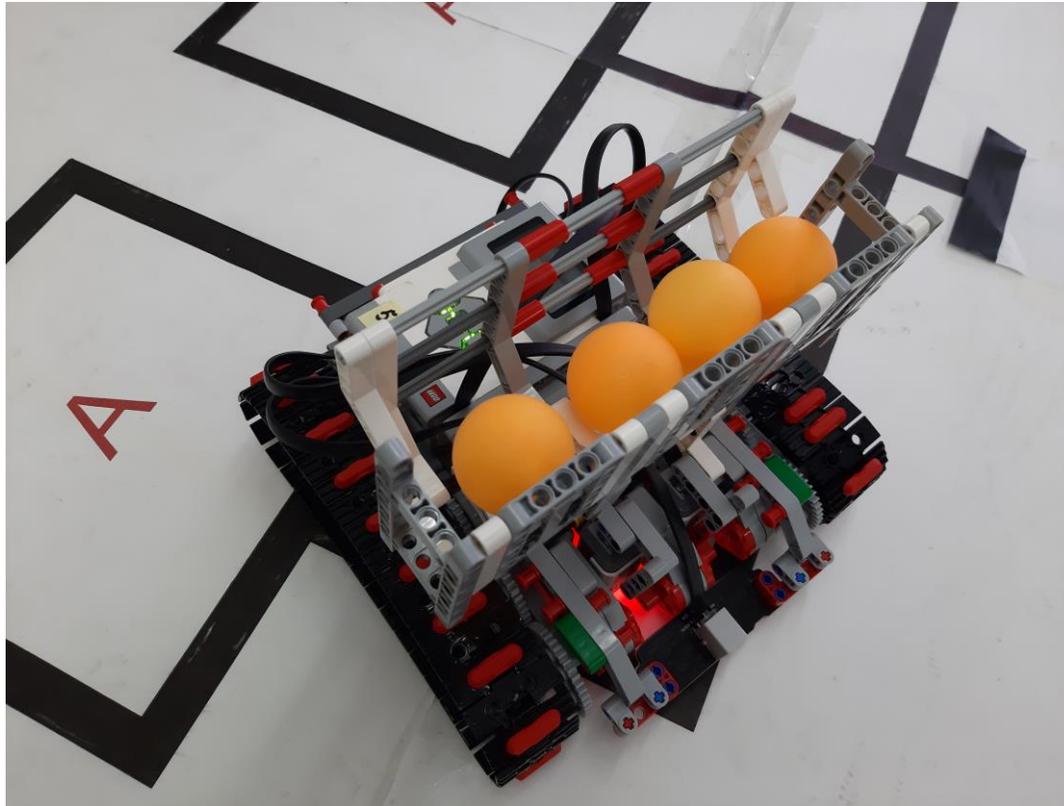
Собрав конструкцию для прохождения по линии и преодоления препятствий, мы приступили к сборке конструкции для «спасения пострадавших».

2. Разработать конструкцию робота для обнаружения и эвакуации пострадавших.

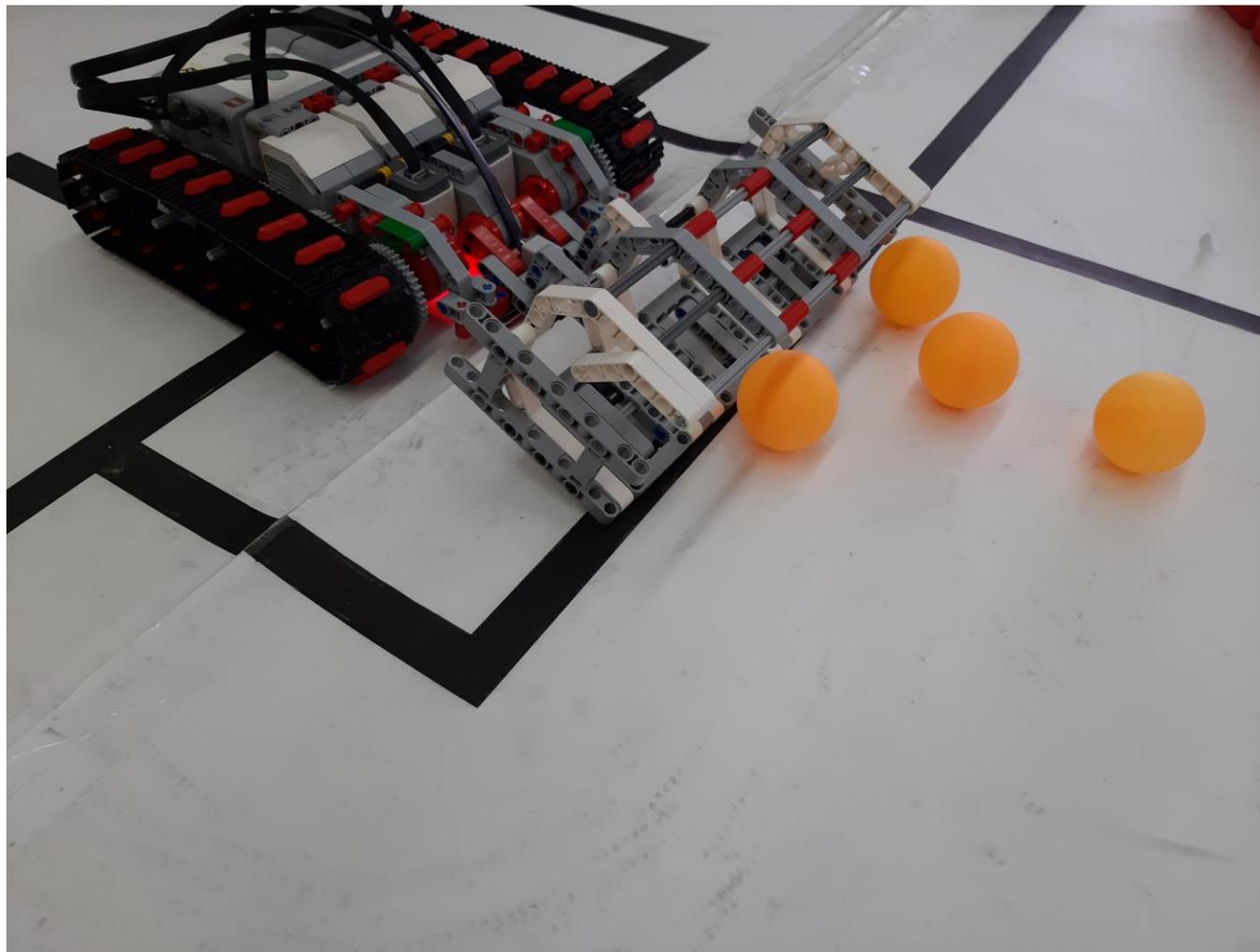
Для сбора пострадавших, мы собрали механизм подъема.



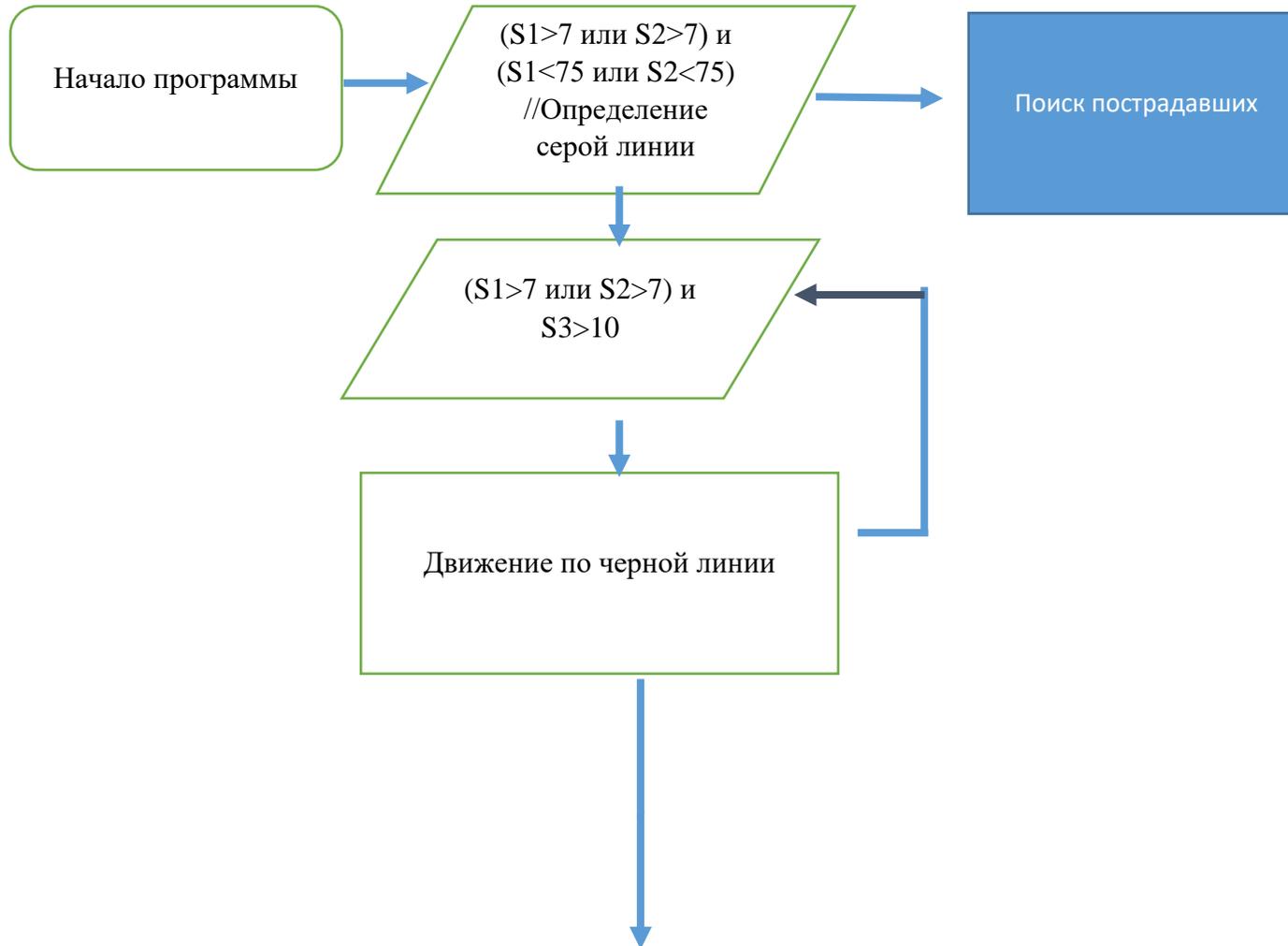
На конструкции робота было решено разместить зону сбора пострадавших.

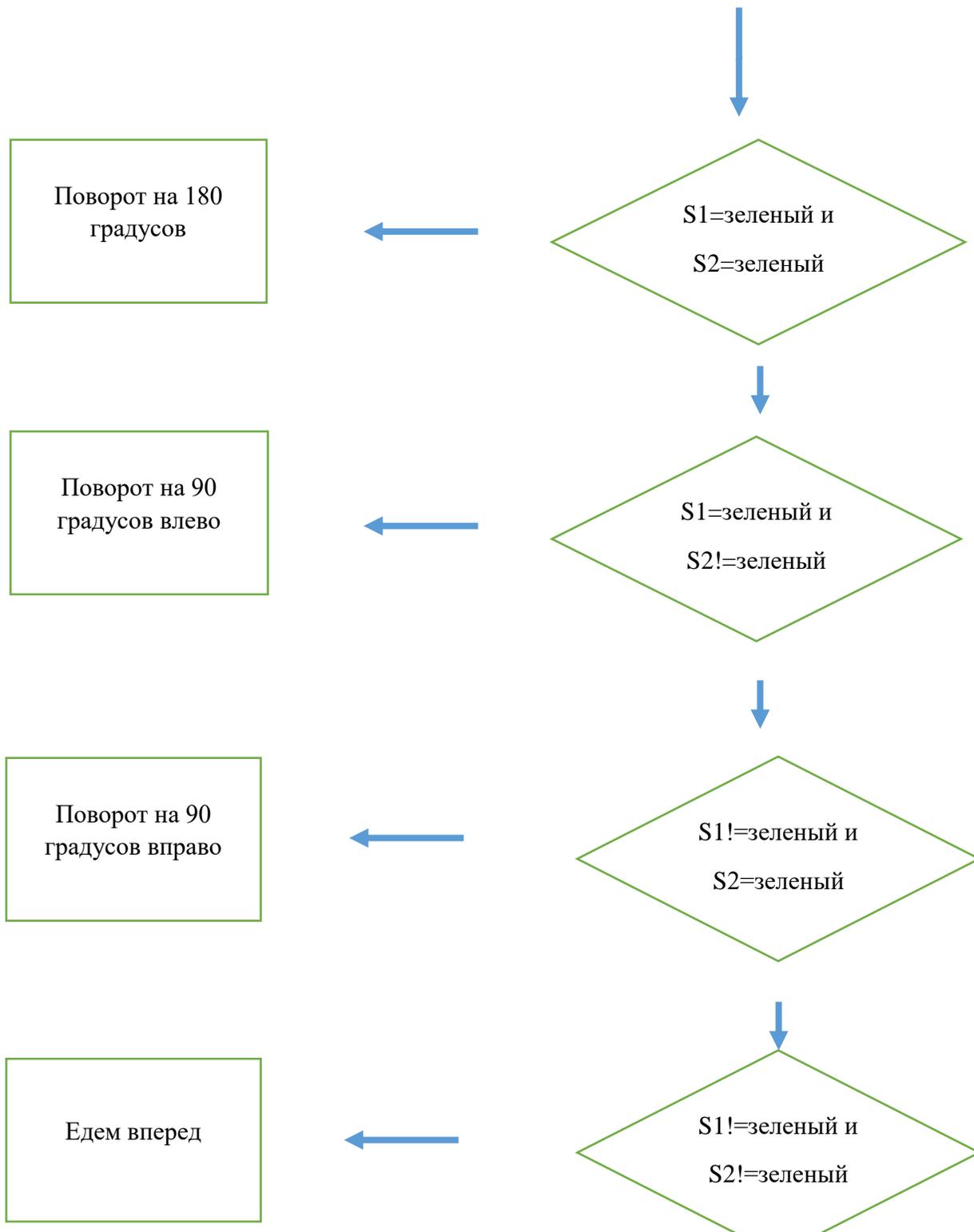


При помощи среднего сервопривода, мы осуществляем эвакуацию пострадавших.



БЛОК-СХЕМА ПРОГРАММЫ





КОД ПРОГРАММЫ

Для езды по линии мы написали PID в среде разработки программ robots.

```
122 void goToLine(int minSpeed, float P, float I, float D )
123 {
124     float diff = SensorValue[S1] - SensorValue[S2];
125     static int oldDiff;
126     static int sum = 0;
127     int dSpeed = (int)(diff * P + sum * I - (oldDiff - diff) * D);
128
129     motor[A] = minSpeed + dSpeed;
130     motor[B] = minSpeed - dSpeed;
131
132     oldDiff = diff;
133     sum += diff;
134 }
135
```

Но значение отраженного света для зеленого и черного цвета почти равны, из-за чего робот сбивался с заданного маршрута. И нам пришлось делать программу для езды по цветам.

```
66 void kek(int I, int gr)
67 {
68     if(getColorName(S1) !=1 && getColorName(S2) !=1)
69     {
70         motor[A]=I;
71         motor[B]=I;
72     }
73     else if(getColorName(S1) !=1 && getColorName(S2) ==1)
74     {
75
76         GOLeft(I, gr);
77     }
78     else if(getColorName(S1) ==1 && getColorName(S2) !=1)
79     {
80         GORight(I, gr);
81     }
82     wait1Msec(1);
83 }
```

Для считывания зеленых меток была применена программа

```
111 wait1Msec(100);
112 Back(15, 39);
113 motor[A]=0;
114 motor[B]=0;
115 wait1Msec(100);
116 while(getColorName(S2) !=1)
117     {
118         motor[A]=-20;
119         motor[B]=20;
120     }
121     motor[A]=0;
122     motor[B]=0;
123 GOLeft(20, 50);
124 wait1Msec(100);
125 c2=getColorName(S2);
126 wait1Msec(1000);
127 c2=getColorName(S2);
128 wait1Msec(400);
129 c2=getColorName(S2);
130 wait1Msec(200);
131 wait1Msec(100);
```

Еще написана программа для обнаружения и объезда препятствий в виде кирпича.

```
91     if (getTouchValue (S3) == 1)
92     {
93         Back (20, 40) ;
94         GOLeft (40, 270) ;
95         GO (30, 340) ;
96         GORight (40, 260) ;
97         GO (30, 740) ;
98         GORight (40, 260) ;
99         GO (30, 345) ;
100        GORight (40, 240) ;
101    }
102    wait1Msec (1) ;
```

Программа для сбора пострадавших:

```
void postr ()
{
    GO (30, 400);
    GORight (30, 660);
    yprCt (-30, 1000);
    GOt (-100, 3000);
    yprCt (50, 1000);
    if (getColorName (S4) !=0)
    {
        GORight (30, 330);
        GO (30, 40);
        Back (30, 40);
        yprCt (-30, 1000);
        GOt (-100, 3000);
        yprCt (50, 1000);
    }

    if (getColorName (S4) !=0)
    {
        GORight (30, 330);
        GO (30, 40);
        Back (30, 40);
        yprCt (-30, 1000);
        GOt (-100, 3000);
    }
}
```