RoboCup RUSSIA OPEN 2024

Президентский физико-математический лицей № 239

RoboCupJunior Rescue Line

Команда: Жужа 7.0

Team Description paper

Состав команды:
Кисанов Владислав
Токарев Матвей

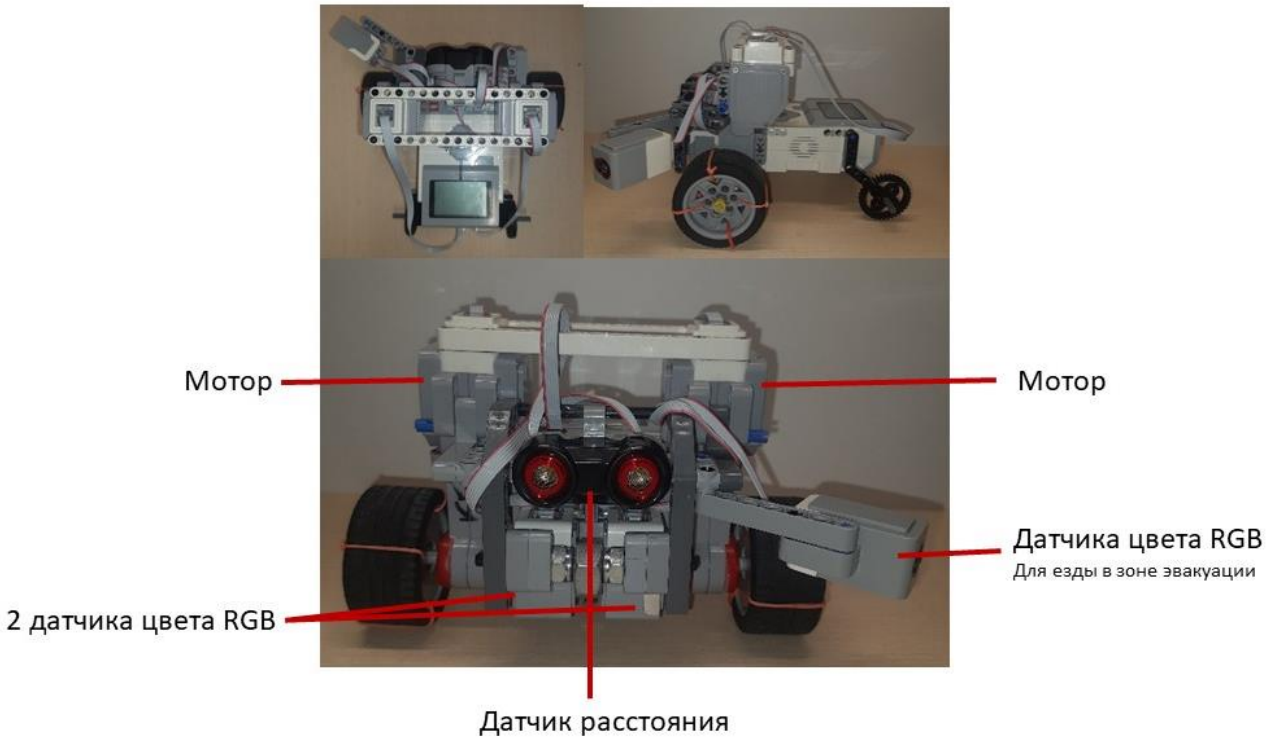
Руководитель:
Моногаров Евгений Владимирович

Организация:
Президентский физико-математический лицей № 239
Санкт-Петербург, Россия

Санкт-Петербург
2024

1. КОНСТРУКЦИЯ

Нами сконструирован робот, который должен самостоятельно выполнить спасательную миссию. Роботу предстоит двигаться по линии через разрушенные препятствия, возвышенности, неровности, чтобы забрать пострадавших и вернуть их на базу, где им будет оказана помощь.



Робот построен на базе Lego MINDSTORMS EV3. В конструкции робота следующие элементы:

- 2 мотора – служат для движения робота

- 1 датчик расстояния – используется для объезда кирпича

- 1 датчик цвета RGB - нужен для езды вдоль стены в зоне эвакуации, расположен под углом 45%

- 2 датчика цвета RGB - предназначен для определения линии, красного, зеленого, серебряного цветов.

2. КОМАНДА

Кисанов Владислав

Капитан, главный программист, конструктор

Токарев Матвей

Программист и конструктор

3. АЛГОРИТМЫ В ПРОГРАММЕ

3.1. Регуляторы линии

3.1.1. Управление моторами

Управляющее воздействие (u) и скорость (v) – это переменные для управления моторами через специальный алгоритм. Управляющее воздействие – это отклонение робота от линии.

Моторы повернуты обратной стороной, поэтому в формуле используется отрицательное значение скорости.

motor[motorB] = -v + u;

motor[motorC] = -v – u.

3.1.2. Пропорциональный регулятор

Пропорциональный регулятор помогает следовать по линии без резких движений. Управляющее воздействие при это растет линейно, за счет чего робот плавно едет по линии.

Коэффициент (kp) – переменная отвечающая за резкость поворота.

e = S1N – S2N;

u = e * kp;

3.1.3. Код линии

```
void(line)
{
e = S1N – S2N;
u = e * kp;
motor[motorB] = -v + u;
motor[motorC]= -v - u;
sleep(1);
}
```

3.2. Нормализация

3.2.1. Что такое нормализация

На линии есть пустые участки. На них робот, едущий по пропорциональному регулятору неизбежно будет съезжать вбок, т.к. у датчиков всегда есть погрешность, и один всегда показывает больше другого, поэтому значения каждого датчика сводится к диапазону от 0 до 1 по калиброванным значениям белого и чёрного.

Нормализованное значение вычисляется по формуле:

S1N = (g1 – black1) / (white1 – black1);

g1 – текущее значение датчика,

white1 – максимальное значение датчика,

black1 – минимальное значение датчика,

S1N – нормализованное значение датчика.

Тоже самое проделывается со 2м датчиком.

3.2.2. Код

```
S1N = (g1 – black1) / (white1 – black1);
S2N = (g2 – black1) / (white1 – black1);
```

3.3. Определение цвета

3.3.1. Алгоритм

Все цвета робот определяет по одному алгоритму: если канал нужного цвета на датчике в x раз больше, чем канал противоположного и при этом канал нужного цвета не слишком темный, то это тот цвет.

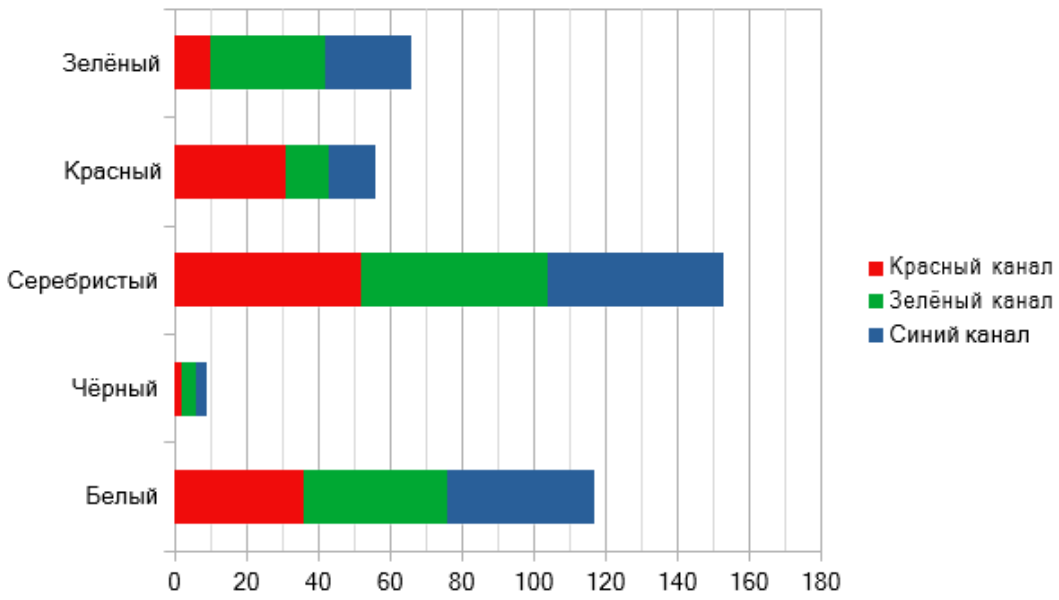
Если значения каждого цвета примерно соответствует откалиброванным значениям (максимальное отклонение занесено в переменную cols), то робот считает, что он на серебряном.

3.3.2. Код зеленого цвета

Красный цвет распознается аналогичным образом.

```
if(g1 > r1 * 2  &&  g1 > 5)
    green1 = true;
else
    green1 = false;
if(g2 > r2 * 2 && g2 > 5)
    green2 = true;
else
    green2 = false;
```

3.3.3. Диаграмма с примером



3.3.4. Код серебряного цвета

```
if(abs(r1 – r1S) < cols && abs(g1 – g1S) < cols && abs(b1 – b1S) < cols)
```

```
    silver1 = true;
else
    silver1 = false;
if(abs(r2 – r2S) < cols && abs(g2 – g2S) < cols && abs(b2 – b2S) < cols)
    silver2 = true;
else
    silver2 = false;
```

3.4.Калибровка

3.4.1. Что такое калибровка?

Калибровка используется для нормализации и получения максимального и минимального значения на линии. Это нужно для нормализации. Также калибровка нужна, чтобы запомнить значения серебряного цвета.

3.4.2. Алгоритм

При запуске калибровки создаются две переменные для каждого датчика: максимальное и минимальное значение этого датчика. Робота двигают по линии, а датчик постоянно опрашивается, и если значение больше максимального, то максимальное значение обновляется, а если ниже минимального — обновляется минимальное значение датчика. Затем робот ставится на серебряный и запоминает свои значения. После ручного завершения калибровки, все значения записываются в файл.

3.4.3. Код

```
int white1, white2, black1, black2;
while(!getButtonPress(buttonDown))
{
    if(g1 > white1)
        white1 = g1;
    if(g1 < black1)
        black1 = g1;
    if(g2 > white2)
        white2 = g2;
    if(g2 < black2)
        black2 = g2;
```

```
    sleep(1);
}
while(!getButtonPress(buttonUp))
    sleep(1);
else
    getColorRGB(S1, r1S, g1S, b1S);
file=fileOpenWrite("kalibrofka.dat");
fileWriteFloat(file, white1);
fileWriteFloat(file, black 1);
fileWriteFloat(file, white2);
fileWriteFloat(file, black 2);
fileWriteLong(file, r1S);
fileWriteLong(file, g1S);
fileWriteLong(file, b1S);
fileWriteLong(file, r2S);
fileWriteLong(file, g2S);
fileWriteLong(file, b2S);
fileClose(file);
```

4. ОГЛАВЛЕНИЕ

1. Конструкция....2
2. Команда....3
3. Алгоритмы в программе....3
3.1. Регуляторы линии....3
3.2. Нормализация....4
3.3. Определение цвета....5
3.4. Калибровка....6
4. Оглавление....8