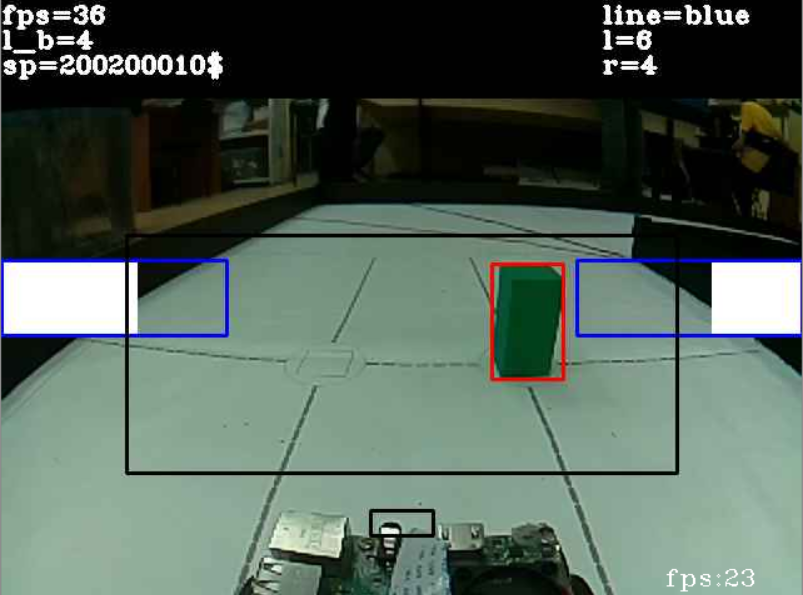
**Реализация системы распознавания препятствий**

Для того чтобы робот мог выполнять все поставленные задачи, мы написали программу на языке python. Разработку мы вели в среде Pycharm. Мы создали на изображении камеры 4 области интереса, изображенных и подписанных на изображении ниже, каждая из которых отвечает за отдельную задачу нашего алгоритма. Тем самым его можно разделить на несколько пунктов: движение по чёрным бортам, поворот, объезд знаков и финиш. Каждый пункт рассмотрен более подробно ниже.





**1. Движение по черным бортам**

Движение по бортам осуществляется с помощью областей интереса №1, №2 и функций **pd()**, **drz()** и **dlz()**. В последних двух наши области интереса разбиваются на 20 частей и на каждом кусочке находим черный. Для каждого датчика существует массив в котором записывается либо **1** если есть черный контур в данной области или **0** если его нет. Самая крайняя единица в массиве является краем контура черного, индекс которой, обозначенный как **i**, в массиве и равен значению нашего датчика.(В левом датчике находим индекс крайней правой единицы и значение датчика равно **i**, для правого датчика находим индекс левой крайней единицы, и значение датчика равно **19 – i**).

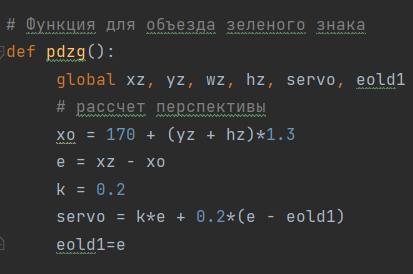
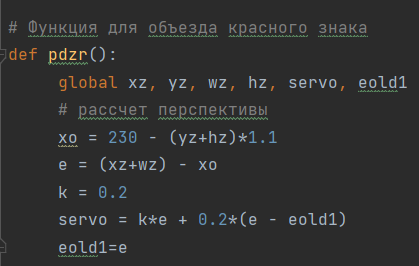
Основной функцией, которая отвечает за само движение, является функция **pd()** – пропорционально-дифференциальный регулятор. В ней мы определяем угол поворота по формуле – **servo = k \* e + \* (e - eold)**, в которой **servo** – значение угла поворота серво мотора, **k** – коэффициент пропорциональности, **e** – ошибка, которая равна разнице между показанием левого и правого датчика (**e = dl - dr**), а **eold** – это ошибка прошлых показаний. Также в функции **pd()** у нас есть ограничение на максимальный угол поворота сервомотора от **-50** до **50** и проверка на случай, при котором на датчиках мы не видим черный цвет. В таком случае мы в зависимости от направления движения поворачиваем на **20** или на **-20**, в зависимости от направления.

**2. Поворот**

Поворот осуществляется благодаря области интереса №3 и функции **dlin()**. Именно в этой области интереса мы находим цвет линии которую мы проезжаем первой, и он будет являться ориентиром в направлении нашего движения. Зная направление мы можем повернуть в нужную сторону, и выполнить другие алгоритмы связанные с направлением движения. Мы также считаем количество пройденных поворотов(линий) и записываем их в переменную **circle**.

**3. Объезд знаков**

В ней мы отслеживаем знаки впереди робота и их цвет. Алгоритм обнаружения знаков реализован в функции **datz()**, в ней мы находим контур увиденного знака, и сохраняем его размеры. В зависимости от цвета увиденного знака, вызывается либо функция **pdzr()**, для красного знака, либо функция **pdzg()**, для зеленого знака.



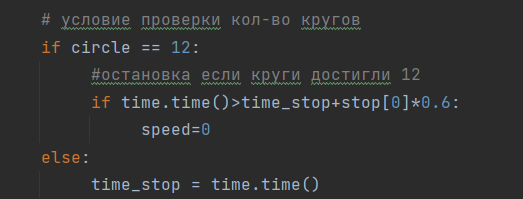
В этих функциях реализован алгоритм объезда знаков. Он работает по принципу пропорционально-дифференциального регулятора, только он заставляет робота ориентироваться не по бортикам, а по контуру знака. В случае красного знака ошибка показаний высчитывается по формуле **e = (xz + wz) – xo**, где **xz** это самая левая координата **x** знака, **wz** – ширина контура, а xo – переменная, в которой учитывается перспектива, относительно координаты **y**, и высчитывается она по формуле **xo = 230 – (yz + hz) \* 1,1** (**yz** – координата **y** нижней точки знака, **hz** – ширина контура знака, а **230** и **1,1** – подобранные коэффициенты). Для зеленого знака ошибка высчитывается как **e = xz – xo**, а учет перспективы идет по формуле **xo = 170 + (yz + hz) \* 1,3**. Таким образом, объезд зеленого знака происходит с левой стороны, а красного справа. Также в случае если робот не видит никакого знака, то он едет по обычному регулятору **pd()**.

**4. Разворот**

После второго круга, в случае если последний кубик до этого был красного цвета, у нас стоит цель развернуться и проехать последний круг в противоположную сторону. Для этого у нас создана функция **razvorot()**. В ней в зависимости от вида разворота, ближе к внешнему или внутреннему бортику, выполняем необходимый алгоритм. Также во время разворота все датчики перестают работать, чтобы не увидеть лишних помех, и не насчитать лишних поворотов.

**5. Финиш**

Финиш робот производит только после проезда 12 перекрестков, то есть когда **circle = 12**, в этом случае мы запоминая до этого время между поворотами, проезжаем половину этого времени и останавливаемся как раз в зоне старта.



**6. Общий алгоритм движения робота.**

***Бесконечный цикл:***

*Считываем изображение*

***Если не выполняется разворот:***

*Считываем линию на кадре*

*Считываем левый бортик на кадре*

*Считываем правый бортик на кадре*

*Считываем знак на кадре*

***Если робот находится в движении:***

***Если не видим знак:***

*Едем по регулятору* ***pd()***

***Если увидели красный знак:***

*Едем по регулятору* ***pdzr()***

***Если увидели зелёный знак:***

*Едем по регулятору* ***pdzg()***

***Иначе:***

*Робот стоит на месте и колеса стоят ровно*

***Если проехали 8 поворотов и последний кубик был красный:***

*Разворот в другую сторону*

***Если проехали 12 поворотов:***

*Проезжаем примерно половину времени между поворотами*

*Останавливаемся*

***Иначе:***

*Запоминаем время*