

Задача 1. «Вышибала»

Робот должен, двигаясь по черной линии с перекрестками, выбить кубики с занимаемых полочек. Кубики имеют размер не менее 30х30х30 мм. Ширина линии 20 мм. Стартовый размер робота не может превышать размеры 200х200 мм. Робот должен полностью уместиться в стартовый квадрат на полигоне. Высота робота не ограничена. На полках могут присутствовать черные кубики, которые нельзя сдвигать. Задание считается выполненным, когда робот выбьет все кубики и остановится в зоне финиша. За каждый выбитый кубик с полки робот получает 10 баллов. За сдвинутый, но не выбитый с полки кубик, робот получает 5 баллов. За сдвинутый черный кубик робот получает штраф 10 баллов. На выполнение задания дается 180 секунд. Если робот выполнил задание полностью, то к баллам добавляется число сэкономленных секунд, разделенное нацело на 10.



(Рис 1. Внешний вид полигона «Вышибала»)

рисунок 1 - задание ВЫШИБАЛА

Задание Робофинист 2023 - ВЫШИБАЛА (рисунок 1)

Для решения задания использовалось 3 блока

- Выезд по старта
- Цикл, из 6-ти повторений для выполнения основного задания
- Заезд на финиш

В первую очередь были написаны блоки езды и поворотов по линии, далее я приступил к написанию логической части того, что и как робот должен выполнять.

В моем случае, я после проезда перекрестка поворачиваю налево, двигаясь с большой скоростью, после некоторого времени замедляю скорость, для точного подъезда к кубику. После остановки на черной линии, всей плашкой ИК датчиков, я считываю отдельным ик датчиком стоящим спереди робота, цвет кубика, после принимаю решение. Либо разворачиваться и двигаться к следующему кубику, либо сбивать кубик посредством опускания балки сервоприводом и резким движением одним из моторов вперед и назад. Далее уже ехать назад, до следующего перекрестка к следующему кубику. И так циклично 6 раз. Блок цикла (рисунок 2). Также представлен фрагмент кода с блоками движения (рисунок 3).

В результате работы, было успешно выполнено задание Вышибала.

```
forward();
for (int i = 0; i < 6; ++i) {
    forward();
    left();
    // delay(1000);
    // right();
    unsigned long int time = millis();
    while (time + 540 > millis()) {
        int P = (bum.getLineAnalog(8) - bum.getLineAnalog(2));
        int z = P * 0.04;
        motor(M - z, M + z);
        bum.setTurnSignal(BUM_TURN_OFF);
    }
    while (bum.getLineAnalog(1) > k && bum.getLineAnalog(9) > k) {
        int P = (bum.getLineAnalog(8) - bum.getLineAnalog(2));
        int z = P * 0.03;
        motor(30 - z, 30 + z);
        bum.setTurnSignal(BUM_TURN_OFF);
    }
    motor(0, 0);
    // delay(1000);
    if (analogRead(A3) > 500) {
        open();
        delay(200);
        motor(100, 0);
        delay(150);
        motor(-100, 0);
        delay(150);
        motor(0, 0);
        close();
        motor(0, 0);
    }
    left();
    forward();
    left();
}
forward();
motor(80, 80);
delay(300);
motor(0, 0);
// forward();
}
```

рисунок 2 - цикл

```

void motor(int l, int r) { // движение
    mot_L.setSpeed( l, MOT_PWM);
    mot_R.setSpeed( r, MOT_PWM);
}
void forward() { // вперед
    while (bum.getLineAnalog (1) > k && bum.getLineAnalog (9) > k ) {
        int P = (bum.getLineAnalog (8) - bum.getLineAnalog (2)) ;
        int z = P * 0.05;
        motor (M - z, M + z);
        bum.setTurnSignal(BUM_TURN_OFF);
    }
    motor (M, M);
    delay (150);
    unsigned long int time = millis();
    while (time + 160 > millis ()) {
        int P = (bum.getLineAnalog (8) - bum.getLineAnalog (2)) ;
        int z = P * 0.05;
        motor (M - z, M + z);
        bum.setTurnSignal(BUM_TURN_OFF);
    }
    motor (0, 0);
}
void right () { //вправо
    while (bum.getLineAnalog (9) < k) {
        motor (Mp, -Mp);
        bum.setTurnSignal(BUM_TURN_RIGHT);
    }
    while (bum.getLineAnalog (9) > k) {
        motor (Mp, -Mp);
        bum.setTurnSignal(BUM_TURN_RIGHT);
    }
    while (bum.getLineAnalog (9) < k) {
        motor (Mp, -Mp);
        bum.setTurnSignal(BUM_TURN_RIGHT);
    }
    motor(0, 0);
}
}

```

рисунок 3 - блок движения