

Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

КУРСОВАЯ РАБОТА

Дисциплина: объектно-ориентированное программирование

Тема: разработка программы езды по черной линии с использованием датчиков света для робота «Omegabot»

Выполнил

Студент группы 3331506/00401

_____ А. А. Тюрин

Преподаватель

_____ М. С. Ананьевский

Санкт-Петербург

2023

Оглавление

| | |
|-------------------------|---|
| Цель..... | 3 |
| Задачи | 3 |
| Реализация..... | 3 |
| Список литературы | 7 |

Цель

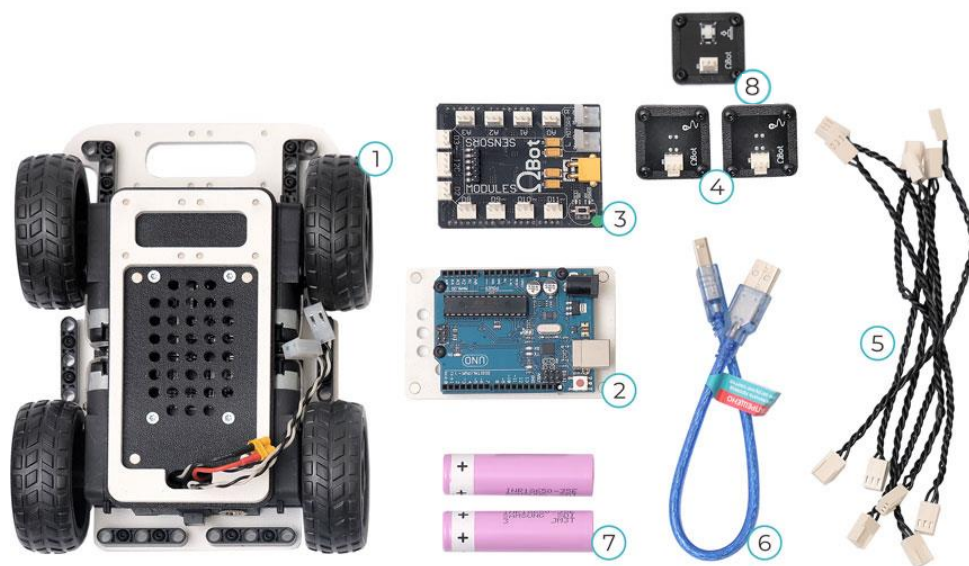
Обеспечить движение по черной линии робота «Omegabot» с использованием датчиков света.

Задачи

- 1) Добиться быстрой скорости ориентирования по линии
- 2) Обеспечить стабильную езду по чёрной линии

Реализация

Для выполнения поставленной задачи была выбрана конфигурация робота, представленная на рисунке 1.



1 - Четырехколесная платформа с базовой основой для установки модулей, 2 – Контроллер, 3 - Плата расширения контроллера, 4 - Модуль датчика линии – 2 шт., 5 - Соединительный провод 3 pin 160 мм, 6 - Кабель для программирования контроллера, 7 – Аккумуляторы, 8 - Модуль кнопка.

Рисунок 1 – Конфигурация робота «Omegabot»

Движение робота начинается только после нажатия кнопки. Остановка двигателей осуществляется повторным нажатием на кнопку.

Движение по линии будет осуществляться последствием снятия с 2ух датчиков света (левого и правого) показаний. Был реализован метод калибровки датчиков для адаптации к освещенности местности. Далее был

реализован пропорционально – дифференциальный (ПД) регулятор, корректирующий скорости, подаваемые на двигатели робота.

Код написан на языке С в программной среде разработки «Arduino».

Реализация описанных методов представлена ниже (Программа 1).

(Пр. 1)

```
#define LEFT_PWM 6
#define LEFT_DIR 7
#define RIGHT_PWM 5
#define RIGHT_DIR 4

#define K_P 5.0 // Proportional gain
#define K_D 20.0 // Differential gain

#define Velocity 190

// Sensor's range
int left_min = 1023;
int left_max = 0;
int right_min = 1023;
int right_max = 0;

int error_old = 0;

bool button = true;
bool button_old = 1;

void drive(const int left_speed, const int right_speed) {
    digitalWrite(LEFT_DIR, left_speed > 0);
    digitalWrite(RIGHT_DIR, right_speed > 0);

    analogWrite(LEFT_PWM, abs(left_speed));
    analogWrite(RIGHT_PWM, abs(right_speed));
}

void stoppage() {
    drive(0, 0);
}

void sensor_calibration(){
    drive(200, -200);
    int left = analogRead(A3);
    int right = analogRead(A2);
    if(left < left_min) left_min = left;
    if(left > left_max) left_max = left;
    if(right < right_min) right_min = right;
    if(right > right_max) right_max = right;
}
```

```

void button_read(bool button, bool button_old) {
    if(digitalRead(A1) == 1 && button_old == 0)
    {
        delay(5);
        if(digitalRead(A1) == 1)
        {
            button = !button;
        }
    }
    button_old = digitalRead(A1);
}

void start_on_button() {
    if(digitalRead(A1) == 1)
    {
        delay(5);
        if(digitalRead(A1) == 1)
        {
            return;
        }
    }
}

void setup() {
    pinMode(LEFT_PWM, OUTPUT);
    pinMode(LEFT_DIR, OUTPUT);
    pinMode(RIGHT_PWM, OUTPUT);
    pinMode(RIGHT_DIR, OUTPUT);
    pinMode(A1, INPUT);
    int tim = millis();

    while(millis() - tim < 4000)
    {
        sensor_calibration();
    }
    stoppage();
    while(1)
    {
        start_on_button();
    }
}

void loop() {
    if(button)
    { // Usage of PD regulator
        int left_sensor = map(analogRead(A3), left_min, left_max, 0, 100);
        int right_sensor = map(analogRead(A2), right_min, right_max, 0, 100);
        double error = left_sensor - right_sensor;
        double u = error * K_P + (error - error_old) * K_D; // Control gain
        drive(constrain(Velocity + u, 250, -100), constrain(Velocity - u, 250, -100));
        error_old = error;
    }
}

```

```
}  
else  
{  
    stoppage();  
}  
button_read(button, button_old);  
}
```

Список литературы

1. Примеры кода: СТАРТ — БАЗОВЫЙ — Omegabot — URL: <https://omegabot.ru/education/examples/base>
2. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. — М.: Горячая линия — Телеком, 2009. С.284 – 387.
3. ПИД регулятор – URL: <https://alexgyver.ru/lessons/pid/>
4. С. Монк. Програмируем Arduino. Профессиональная работа со скетчами. — СПб.: Питер, 2017.