

Санкт-Петербургский политехнический университет Петра Великого  
Институт металлургии, машиностроения и транспорта

# Курсовая работа

Дисциплина: Объектно-ориентированное программирование

Тема: прохождение лабиринта омегаботом

Разработал: Студент гр. 3331506/10401

Косенко А.А.

Преподаватель

Ананьевский М.С.

Санкт-Петербург

2024г.

## Оглавление

Введение .....	3
Датчики навигации .....	4
Постановка задачи .....	6
Код программы .....	8
Результат работы алгоритма .....	11
Заключение .....	11

## Введение

**Omegabot** — это робототехническая платформа с программируемыми модулями, разработанная специально для обучения робототехнике и программированию в визуальной среде, C++ и Python в образовательных учреждениях.

### Особенности платформы:

**Многоуровневая система обучения.** Позволяет детям младшего возраста начать изучать робототехнику и постепенно повышать компетенции.

**Вариативность.** Сменные модули фиксируются на платформе с помощью магнитных креплений, а количество предлагаемых модулей, в зависимости от наборов, позволяет решать разные задачи.

**Универсальность.** Использование в основе программируемого контроллера архитектуры Arduino и открытых универсальных интерфейсов и разъёмов для подключения обеспечивает совместимость конструктора с другими внешними модулями сторонних производителей.

**Безопасность.** Колесная платформа, составляющая конструктивную основу робота, выполнена из АВС-пластика, что обеспечивает более высокую прочность и повышенную безопасность при проведении занятий с детьми.

**Omegabot может быть использован для участия в соревнованиях** (Робофест, Робофинист, классических соревнованиях по робототехнике), а также в олимпиаде НТИ.



Рисунок 1 Робототехническая платформа Омегабот

## Датчики навигации

### Модуль УЗ дальномера.

используя принцип эхолокации, посылает сигнал и принимает его отражение от объекта. Время между отправкой и получением сигнала используется для определения расстояния. В состав ультразвукового дальномера входят дальномер HC-SR04 и вычислительный модуль.

Принцип работы дальномера HC-SR04 заключается в излучении ультразвукового сигнала и приеме его отражения от объекта наблюдения. При этом запуск излучения осуществляется каким-либо внешним устройством управления подачей импульса на вывод, а при приеме отраженного сигнала дальномер HC-SR04 формирует импульс на выводе. Таким образом может быть измерено время между началом излучения и приемом отраженного сигнала. То есть то время, за которое ультразвук преодолел путь, равный двум расстояниям до объекта наблюдения. Результатом умножения половины полученного значения времени на скорость звука является расстояние до объекта наблюдения.

Все перечисленные функции и вычисления автоматически выполняет вычислительный модуль дальномера, построенный на основе восьмиразрядного микроконтроллера. При обращении по интерфейсу I2C он выдает расстояние до объекта наблюдения в сантиметрах в диапазоне от 2 до 254 см.

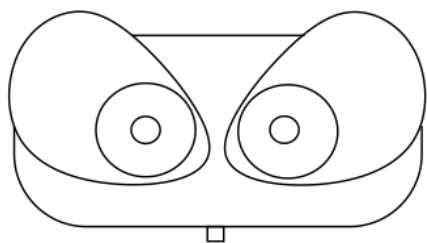


Рисунок 2 Внешний вид модуля УЗ дальномера

### Модуль датчика касания

Модуль датчика касания представляет собой печатную плату с установленным чувствительным элементом – концевым выключателем. Внешний вид модуля датчика касания приведен на рисунке 3

Области применения модуля датчика касания:

- обнаружение препятствия во время движения;
- срабатывание при физическом контакте.

При физическом контакте модуля датчика касания с препятствием во время движения происходит замыкание контактов концевого выключателя, в результате чего вырабатывается соответствующий сигнал.



Рисунок 3 Внешний вид датчика касания

### Модуль датчика линии

Модуль датчика линии представляет собой печатную плату с установленным оптическим датчиком предназначенным для обнаружения препятствий в виде линий (белых или чёрных) на основании определения отражающей способности поверхности. Внешний вид модуля датчика линии приведен на рисунке 2.8. Датчик является аналоговым, выходной сигнал с сенсора прямо пропорционален количеству отраженного от поверхности света.

Области применения модуля датчика линии:

- определение факта наличия предмета известного цвета рядом с модулем;
- дифференциация предметов с разными по цвету поверхностями;
- в Конструкторе используется для перемещения робота по линии.

В модуле датчика линии установлена оптопара TCRT5000, излучающая свет в инфракрасном спектре и принимающая отражённый сигнал. В зависимости от типа и цвета поверхности меняется интенсивность отраженного сигнала. По этой интенсивности можно судить об оттенках цвета поверхности.

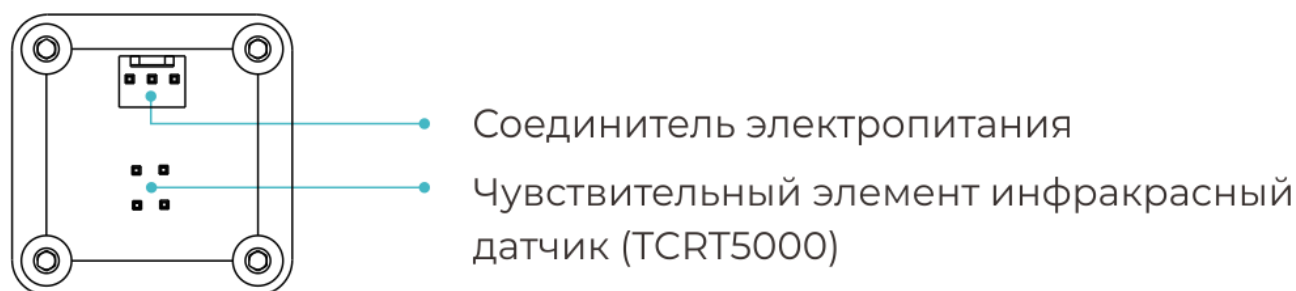


Рисунок 4 Внешний вид модуля датчика линии

### Расширенная платформа датчика расстояния

Расширенная основа представляет собой деталь из АВС пластика с установленными датчиками расстояния. Внешний вид расширенной основы с датчиками расстояния приведен на рисунке 2.7. Некоторые из отверстий имеют встроенные магниты для фиксации модулей. С передней и боковых сторон основы расположены датчики расстояния.

Датчики расстояния предназначены для определения расстояния до объектов и состоят из комбинации устройств: датчика положения, инфракрасного светодиода и схемы обработки сигнала.

Уровень напряжения на выходе датчиков прямо пропорционален расстоянию до объекта. Датчики расстояния возможно использовать в качестве датчиков приближения.

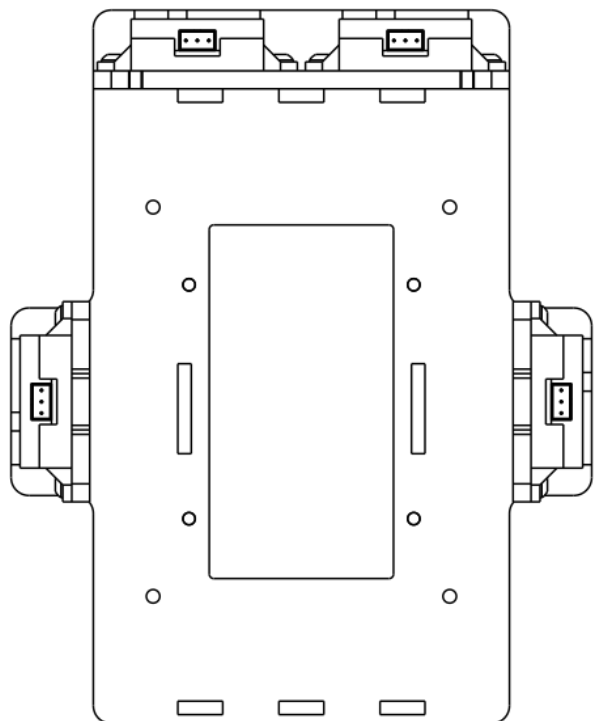


Рисунок 5 Внешний вид расширенной основы с датчиками расстояния

## Постановка задачи

Разработать и реализовать программу для робота Омегабот, позволяющую ему проехать лабиринт, используя ультразвуковые датчики для навигации. Робот должен двигаться по левой стенке лабиринта, избегая столкновения с ней, и достичь цели без ошибок.



поиска стены (State\_FIND).

- В зависимости от целевого состояния перейти в соответствующую функцию состояния:
- forward(): Двигаться вперед с регулируемой скоростью.
- right(): Повернуть на 90 градусов вправо.
- left(): Повернуть на 90 градусов влево.
- find(): Выполнить поведение поиска стены (поворачивать на месте и двигаться вперед, пока не будет обнаружено

## Код программы

```
#define SPEED_LEFT 5
#define SPEED_RIGHT 6
#define DIR_LEFT 4
#define DIR_RIGHT 7
#define LEFT_ULTRASONIC_TRIG_PIN 8
#define LEFT_ULTRASONIC_ECHO_PIN 9
#define FRONT_ULTRASONIC_TRIG_PIN 10
#define FRONT_ULTRASONIC_ECHO_PIN 11
#define SOUND_PIN 12
```

```
#define State_FORWARD 0
#define State_RIGHT 1
#define State_LEFT 2
#define State_FIND 3
```

```
#define MAX_ROTATION_SPEED 250
#define MAX_MOVE_SPEED 180
#define MIN_ROTATION_SPEED 200
#define MIN_MOVE_SPEED 150
#define SPEED_STEP 2
```

```
int current_speed = 0;
int State, TargetState;
unsigned long timer;
```

```
void move(bool left_direction, bool right_direction, uint8_t left_speed, uint8_t right_speed) {
  analogWrite(SPEED_LEFT, left_speed);
  analogWrite(SPEED_RIGHT, right_speed);
  digitalWrite(DIR_LEFT, left_direction);
  digitalWrite(DIR_RIGHT, right_direction);
}
```

```
void forward() {
  digitalWrite(SOUND_PIN, 0);
  timer = millis();
  if (State == State_FORWARD && current_speed + SPEED_STEP >= MAX_MOVE_SPEED)
    current_speed = MAX_MOVE_SPEED;
```



```

else if (State == State_FORWARD)
current_speed += SPEED_STEP;
else {
State = State_FORWARD;
current_speed = MIN_MOVE_SPEED;
}
move(1, 1, current_speed, current_speed);
}

void right() {
State = State_RIGHT;
timer = millis();
digitalWrite(SOUND_PIN, 0);
move(1, 0, MAX_ROTATION_SPEED, MIN_ROTATION_SPEED);
delay(1000); // Adjust delay to achieve 90 degrees rotation
move(1, 1, 0, 0); // Stop after rotation
}

void left() {
State = State_LEFT;
timer = millis();
digitalWrite(SOUND_PIN, 0);
move(0, 1, MIN_ROTATION_SPEED, MAX_ROTATION_SPEED);
delay(1000); // Adjust delay to achieve 90 degrees rotation
move(1, 1, 0, 0); // Stop after rotation
}

void find() {
if (State != State_FIND)
timer = millis();
State = State_FIND;
if ((millis() - timer) > 5000) {
digitalWrite(SOUND_PIN, 1);
move(1, 1, 0, 0);
} else {
move(1, 1, 100, 100);
delay(100);
move(1, 0, 100, 100);
delay(100);
}
}

long readUltrasonicDistance(int trigPin, int echoPin) {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

```

```

long duration = pulseIn(echoPin, HIGH);
return duration * 0.034 / 2;
}

void setup() {
  Serial.begin(9600);
  pinMode(LEFT_ULTRASONIC_TRIG_PIN, OUTPUT);
  pinMode(LEFT_ULTRASONIC_ECHO_PIN, INPUT);
  pinMode(FRONT_ULTRASONIC_TRIG_PIN, OUTPUT);
  pinMode(FRONT_ULTRASONIC_ECHO_PIN, INPUT);
  pinMode(SPEED_LEFT, OUTPUT);
  pinMode(SPEED_RIGHT, OUTPUT);
  pinMode(DIR_LEFT, OUTPUT);
  pinMode(DIR_RIGHT, OUTPUT);
  pinMode(SOUND_PIN, OUTPUT);
}

void loop() {
  long left_distance = readUltrasonicDistance(LEFT_ULTRASONIC_TRIG_PIN,
  LEFT_ULTRASONIC_ECHO_PIN);
  long front_distance = readUltrasonicDistance(FRONT_ULTRASONIC_TRIG_PIN,
  FRONT_ULTRASONIC_ECHO_PIN);

  if (front_distance > 20 && left_distance < 15) {
    TargetState = State_FORWARD;
  } else if (front_distance == 15) {
    TargetState = State_LEFT;
  } else if (front_distance > 20 && left_distance > 15) {
    // No obstacles detected, robot should "find" the wall
    TargetState = State_FIND;
  }

  switch (TargetState) {
    case State_FORWARD:
      forward();
      break;
    case State_RIGHT:
      right();
      break;
    case State_LEFT:
      left();
      break;
    case State_FIND:
      find(); // Perform wall finding behavior
      break;
  }
}

```

## Результат работы алгоритма

Робот омегабот успешно проехал лабиринт, используя ультразвуковые датчики и не касаясь левой стенки. Алгоритм, разработанный для этой задачи, позволил роботу без проблем следовать по маршруту и избегать столкновений с препятствиями. Результат работы алгоритма был положительным, и нам удалось достичь поставленной цели.

## Заключение

В ходе выполнения курсового проекта были разработаны алгоритмы движения и принятия решений, позволяющие роботу успешно проходить лабиринт, избегая препятствия.

Этот проект позволил не только углубить знания в области робототехники и программирования, но также на практике оценить работу ультразвуковых датчиков и их роль в автономном движении роботов. Полученные результаты позволяют сделать вывод о возможности успешного применения данной технологии в различных сферах, где требуется автономное движение роботов.

Выполнение данного курсового проекта позволило не только приобрести новые знания и навыки, но также применить их на практике, что было ценным опытом для дальнейшего профессионального роста в области робототехники и программирования.