Министерство просвещения РФ

Гимназия №1505

Пояснительная записка к проектной работе

«Vexour робот»

Выполнили ученики 6 «В» класса

Бровкин Иван, Шпак Артем

Консультант проекта:

Коняхин А.В.

Москва

2018

# Актуальность проекта

В современном мире мы всё чаще начинаем использовать роботов для выполнения трудно и опасной для жизни человека работы. Чтобы собрать робота, который сможет выполнить сложные и ответственные функции необходимо много учиться. Для ознакомления с робототехникой существуют множество различных конструкторов. Самый популярный среди детей и молодежи, конечно, же Lego (Лего). Но сейчас набирают популярность конструкторы на платформе Arduino. Одним из таких конструкторов является набор Vex или его российский аналог от компании «Технолаб». Для нашей школы это совершенно новый конструктор. В нашем проекте мы разбирались, как собрать механизм робота и запрограммировать его на выполнения данных функций, чтобы в дальнейшем можно было использовать полученные результаты на курсах по робототехнике, которые проходят в нашей школе.

# Цель проекта

Приобретение необходимых компетенций для подготовки к соревнованиям «МОСКОВСКИЕ МАСТЕРА» по направлению робототехника.

#  Задачи проекта

1. Изучить основные принципы конструирования роботов.
2. Определить основные функции, которыми должен обладать робот для участия в соревнованиях. Узнать, как реализовать их с помощью датчиков и программирования.
3. Собрать робота (с автономным управлением), обладающего функциями схвата и движения по заданной траектории.
4. Написать программы для управления роботом: «Объезд препятствия», «Обнаружение препятствия и перемещение его на свободное место», «Движение по заданной траектории».
5. Создать видеоролик в котором мы поможем преодолеть сложные моменты при сборке или программировании робота.
6. Создать тренировочную площадку.
7. Провести испытания робота.

# Характеристики робота

Собранный нами робот обладает следующими характеристиками:

1. Питание осуществляется от аккумуляторной батареи 7,2В.
2. Робот оснащен 3 датчиками: ультразвуковым дальномером и два ИК-датчика.
3. Робот собран из деталей, входящих в набор «Технолаб» и его габаритные размеры (ВхШхГ) не превышают 450x450x450 мм.
4. Управление робота – автономное: программа управления написан на языке Arduino.

# Описание работы робота

Представленный робот (рис. 1) управляется от аппаратной платформы «Технолаб» (аналог Arduino Mega 256). Платформа 1 содержит аналоговые и цифровые порты, а также силовые порты для управления движением механизмов робота.

Движение робота осуществляется моторами 2, которые подключаются к силовым портам М1 и М2 платформы. Программа управления моторами описана в библиотеке «Motor» (см приложение 4). За управление движением руки робота 3 отвечает сервопривод 4 (порт 5), а за управление схватом 6 – сервопривод 5 (порт 2).



Рис. 1. Механизмы робота

Обнаружение препятствия осуществляется с помощью ультразвукового датчика (7), подключаемого к портам (37, 36). Датчик определяет препятствие в радиусе 400 мм. Программа работы ультразвукового датчика описана в библиотеке «Sonar» (см приложении 5). Датчик установлен на передней панели робота. !Внимание: чтобы датчик правильно реагировал необходимо следить, чтобы перед ним не было посторонних предметов, например, проводов от ИК-датчиков.

Движение по заданной траектории (изменение траектории движения) осуществляется с помощью инфракрасных датчиков (8), подключаемые к портам (35, 34) Особенность данных датчиков в том, что они могут реагировать только на расстоянии не более 10 мм от пола. Для правильной работы пришлось дооснастить наш робот специальными устройствами, удерживающими датчики на заданном расстоянии. Датчики установлены справа и слева от звукового дальномера. С помощью ИК-датчиков осуществляется реализация программы «Движение по черной линии» (см Приложение 2). В данном проекте показано изменение траектории движения робота при наезде на черную линию.

# Трудности при выполнения проекта

При решении задачи мы столкнулись с некоторыми трудностями:

1. Пришлось с «нуля» изучат язык программирования для аппаратной платформы.
2. Не очень удобно проводить экспериментальные испытания робота из-за невозможности записать несколько программах кодов одновременно на плату, а также из-за небольшой емкости аккумуляторной батареи (требуется частая подзарядка)
3. Не удалось отрегулировать скорость движения робота, из-за чего не удалось осуществить движение робота по черной линии (робобиатлон).

# Результаты проекта.

Мы решили основные задачи нашего проекта: собрали робота, обладающего функцией схвата, научили его двигаться и выполнять определенные функции. С помощью разработанных программ можно осуществлять: захват объекта, перенос объекта на другую позицию и движение по черной линии (листинги всех программ представлены в Приложениях 1-5). С работой робота можете ознакомиться в прилагаемом к пояснительной записке видео. Также мы записали видео, в котором рассказываем, как написать программу управления движением робота на языке Arduino версии 1.8.7.

Для участия в соревновании «Московские мастера» робот должен уметь: перемещаться прямо на 1 м вперед-назад и выполнять поворот направо и налево на 90 градусов. Собранный робот с данными задачами справляется.

Используя разработанные программы, любой ученик нашей школы, занимающийся на спецкурс «Робототехника» (и не только), сможет запрограммировать робота, работающего на платформе Arduino и его аналогах, на выполнение описанных функций, которые могут использоваться, например, для прохождения лабиринта или движения робота внутри черного круга, а также других заданий.

# Перспективы развития проекта.

Программирование роботов на платформе Arduino открывают широкие возможности для реализации самых разнообразных задач. В последствии мы планируем расширить функционал нашего робота и научить его распознавать объекты по цвету (белый/черный).

Приложение №1

**Листинг программы «Объезд препятствия»**

//программа объезда препятствия

#include "motor.h"

#include "sonar.h"

unsigned int dinst;

void setup() {

 setup\_motor\_system(46,45,6,7);

\_stop;

Sonar\_init(37,36);

Serial.begin(9600);

}

void loop() {

 //считываем показания датчиков

 Serial.println (dinst);

dinst=Sonar(400);

//если дистанция до препятствия менее 30 мм, то выполняется программа

 if (dinst<=30){

\_stop();

left();//поворот налево

delay(500);

forward();//вперед

delay(500);

\_stop();//остановка

delay(500);

right();//поворот направо

delay(500);

forward();//вперед

delay(1500);

\_stop();//остановка

delay(500);

right();//поворот направо

delay(500);

forward();//вперед

delay(500);

\_stop();//остановка

delay(500);

left();//поворот налево

delay(500);

\_stop();

delay(500);

}

//основное движение робота

else{

 forward();//вперед

}

 }

Приложение №2

**Листинг программы «Движение по черной линии и очистка траектории от препятствия»**

#include<Servo.h>

#include "motor.h"

#include "sonar.h"

Servo grab;

Servo roll;

unsigned int dinst;

//порты для ИК-датчков

const int L\_sensor=35;

const int R\_sensor=34;

//переменные ИК-датчиков

int l;

int r;

void setup() {

 pinMode(L\_sensor, INPUT);

 pinMode(R\_sensor, INPUT);

grab.attach(2);

roll.attach (5);

setup\_motor\_system(46,45,6,7);

\_stop;

\_time=millis();

Sonar\_init(37,36);

Serial.begin(9600);

}

void loop() {

 //считываем показания датчиков

 Serial.println (dinst);

l = digitalRead(L\_sensor);

r = digitalRead(R\_sensor);

dinst=Sonar(400);

//если дистанция до препятствия менее 30 мм, то выполняется программа

 if (dinst<=30){

\_stop();

reveal();// раскрыть схват

delay(500);

toclose();//сомкнуть схват

delay (2000);

reroll\_down();//поднял руку

delay (1800);

left\_l();//поворот налево на месте на 90 град

delay(900);

\_stop();//остановка

delay(900);

forward();//вперед

delay(900);

\_stop();

delay(900);

reroll\_up();//опустил руку

delay(2000);

reveal();// раскрыл схват

delay(2000);

reverse();//назад

delay(500);

\_stop();//остановка

delay(900);

right\_l();//поворот направо на месте 90град

delay(900);

\_stop();//остановка

delay(900);

}

//основное движение робота

else{

 //если датчик считывает белый фон, то едет прямо

 if ((l == 0)&&(r==0)) {

forward();//вперед

 }

 //если один датчик считывает черный фон, то поворачивает

 if ((l==1)&& (r==0)){

 left();

 delayMicroseconds(10);

 }

if ((l==0)&& (r==1)){

 right();

 delayMicroseconds(10);

}

//если оба датчика считывают черный фон, то остановка

if ((l==1)&& (r==1)){

 \_stop();

}

 }

}

 //команды работы схвата

void reveal(){

 grab.writeMicroseconds(1800);

 delay(900);

 grab.writeMicroseconds(1500);

}

void toclose(){

 grab.writeMicroseconds(1200);

 delay(900);

 grab.writeMicroseconds(1500);

}

void reroll\_up(){

 roll.writeMicroseconds(1800);

 delay(1000);

 roll.writeMicroseconds(1500);

}

void reroll\_down(){

 roll.writeMicroseconds(1200);

 delay(1000);

 roll.writeMicroseconds(1500);

}

Приложение №3

**Листинг программы «Обнаружение препятствия, захват и перемещение на другую позицию»**

#include<Servo.h>

#include "motor.h"

#include "sonar.h"

Servo grab;

Servo roll;

unsigned int dinst;

void setup() {

grab.attach(2);

roll.attach (5);

setup\_motor\_system(46,45,6,7);

\_stop;

Sonar\_init (37,36);

Serial.begin(9600);

}

void loop() {

 //считываем показания датчиков

 Serial.println (dinst);

dinst=Sonar(400);

//если дистанция до препятствия менее 30 мм, то выполняется программа

 if (dinst<=30){

\_stop();

reveal();// раскрыть схват

delay(500);

toclose();//сомкнуть схват

delay (2000);

reroll\_down();//поднял руку

delay (1800);

left\_l();//поворот налево на месте на 90 град

delay(900);

\_stop();//остановка

delay(900);

forward();//вперед

delay(900);

\_stop();

delay(900);

reroll\_up();//опустил руку

delay(2000);

reveal();// раскрыл схват

delay(2000);

reverse();//назад

delay(500);

\_stop();//остановка

delay(900);

right\_l();//поворот направо на месте 90град

delay(900);

\_stop();//остановка

delay(900);

}

//основное движение робота

else{

forward();//вперед

 }

 //команды работы схвата

void reveal(){

 grab.writeMicroseconds(1800);

 delay(900);

 grab.writeMicroseconds(1500);

}

void toclose(){

 grab.writeMicroseconds(1200);

 delay(900);

 grab.writeMicroseconds(1500);

}

void reroll\_up(){

 roll.writeMicroseconds(1800);

 delay(1000);

 roll.writeMicroseconds(1500);

}

void reroll\_down(){

 roll.writeMicroseconds(1200);

 delay(1000);

 roll.writeMicroseconds(1500);

}

Приложение №4

**Библиотека «Движение робота»**

/левый двигатель порт M1

int motor\_L1;

int motor\_L2;

//правый двигатель порт M2

int motor\_R1;

int motor\_R2;

void setup\_motor\_system(int L1, int L2, int R1, int R2)

{

motor\_L1=L1;

motor\_L2=L2;

motor\_R1=R1;

motor\_R2=R2;

pinMode(motor\_L1, OUTPUT);

pinMode(motor\_L2, OUTPUT);

pinMode(motor\_R1, OUTPUT);

pinMode(motor\_R2, OUTPUT);

Serial.begin(9600);

}

void forward (){

//движение вперед

digitalWrite (motor\_L1,LOW);

digitalWrite (motor\_L2, 24);

digitalWrite (motor\_R1, LOW);

digitalWrite (motor\_R2, 24);

}

void right (){

//направо

digitalWrite (motor\_L1, 24);

digitalWrite (motor\_L2, LOW);

digitalWrite (motor\_R1, LOW);

digitalWrite (motor\_R2, 24);

}

void right\_l (){

//направо на месте

digitalWrite (motor\_L1, LOW);

digitalWrite (motor\_L2, LOW);

digitalWrite (motor\_R1, 24);

digitalWrite (motor\_R2, LOW);

}

void left (){

//налево

digitalWrite (motor\_L1, LOW);

digitalWrite (motor\_L2, 24);

digitalWrite (motor\_R1, 24);

digitalWrite (motor\_R2, LOW);

}

void left\_l (){

//налево на месте

digitalWrite (motor\_L1, 24);

digitalWrite (motor\_L2, LOW);

digitalWrite (motor\_R1, LOW);

digitalWrite (motor\_R2, LOW);

}

void reverse (){

//назад

digitalWrite (motor\_R2, LOW);

digitalWrite (motor\_R1, 24);

digitalWrite (motor\_L2, LOW);

digitalWrite (motor\_L1, 24);

}

void \_stop (){

//остановка

digitalWrite (motor\_L1, LOW);

digitalWrite (motor\_L2, LOW);

digitalWrite (motor\_R1, LOW);

digitalWrite (motor\_R2, LOW);

}

Приложение №5

**Библиотека «Ультразвуковой датчик препятствия»**

//бибилиотека для ультразвукового датчика

int Trig;

int Echo;

Sonar\_init(int Tr, int Ech)

{

 Trig=Tr;

 Echo=Ech;

 pinMode(Trig, OUTPUT);

 pinMode (Echo, INPUT);

}

//измерение расстояния

int Sonar(unsigned long Limit)

{

 int Long\_cm;

 unsigned long Lim=Limit\*58;

 digitalWrite(Trig, LOW);

 delayMicroseconds(2);

 digitalWrite(Trig, HIGH);

 delayMicroseconds(10);

 digitalWrite(Trig, LOW);

 Long\_cm=int(pulseIn(Echo,HIGH,Lim)/58);

 if(Long\_cm==0) return int(Limit);

 return Long\_cm;

}