НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра «Вычислительные системы и технологии»

ОТЧЕТ

*по лабораторной работе №1*

Дисциплина: **«**Аппаратное и программное обеспечение   
роботизированных систем**»**

Выполнил(а):

Студент гр. 17-В-1

Савин Д. В.

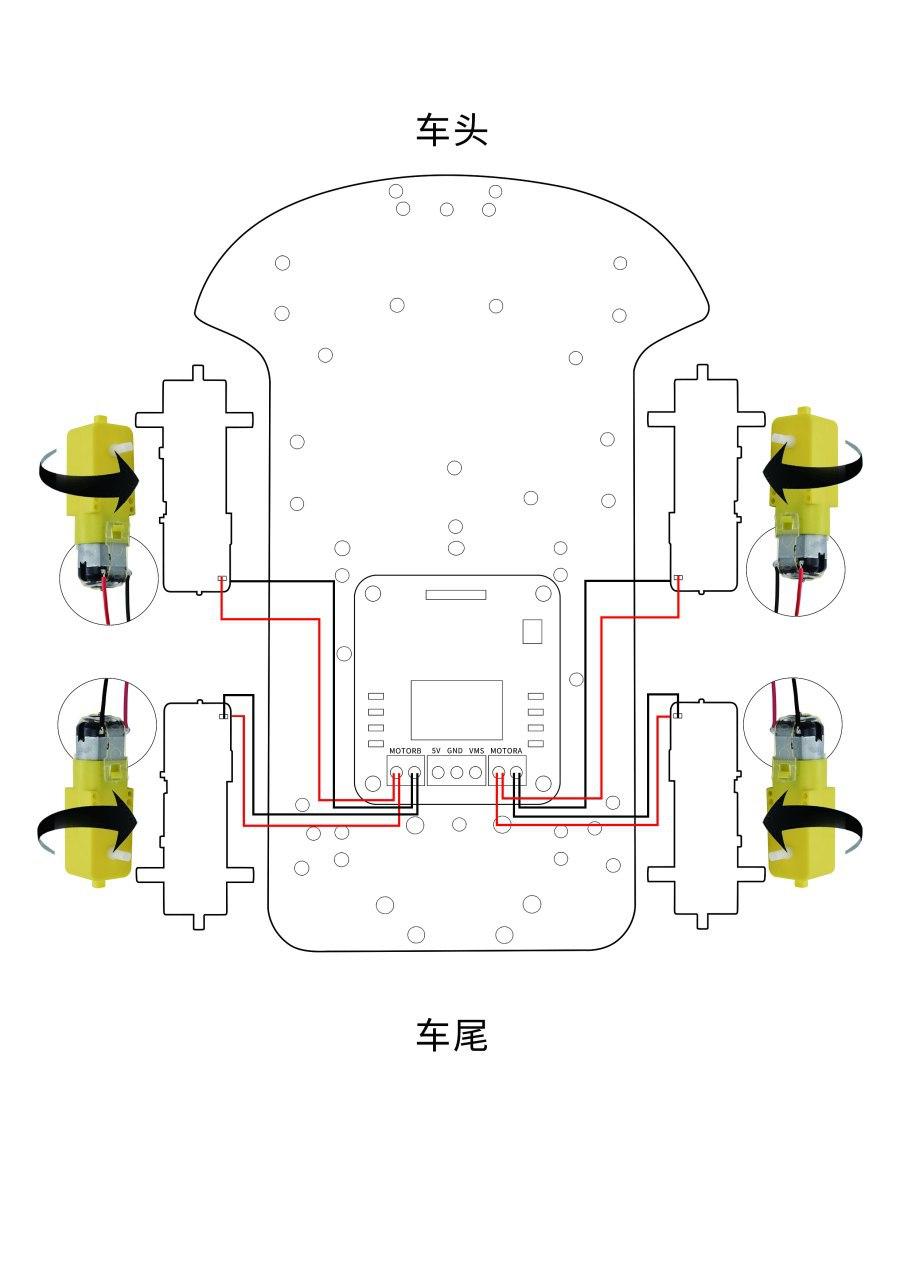
Проверил:

Гай В. Е.

Нижний Новгород

2020

**1. Схема робота**

****

**2. Code for robot**

#include <IRremote.h> // подключение библиотеки (Требуется для работы Ардуино с ИК приемником)

int receiverpin = 11;// определение вывода инфракрасного сигнала

int pinI1=A1;// определение I1 интерфейса

int pinI2=A0;// определение I2 интерфейса

int speedpin1=6;// определение EA интерфейса (управление скоростью)

int pinI3=A3;// определение I3 интерфейса

int pinI4=A2;// определение I4 интерфейса

int speedpin2=5;// определение EB интерфейса (управление скоростью)

#define ADVAN 0xFF629D // пульт дистанционного управления (клавиша 7 «вперед») #define BAC 0xFFA857 // пульт дистанционного управления (клавиша 8 «назад»)

#define STO 0xFF02FD // пульт дистанционного управления (клавиша 6 «стоп»)

#define left 0xFF22DD // пульт дистанционного управления (клавиша 5)

#define right 0xFFC23D // пульт дистанционного управления + ключ 4

IRrecv irrecv(receiverpin);//инициализируем порт 11 как порт ИК приемника

// Загрузив программу, Arduino дает нашему коду возможность поучаствовать в инициализации системы. Для этого мы должны указать микроконтроллеру команды, которые он выполнит в момент загрузки и потом забудет про них (т.е. эти команды выполнятся только один раз при старте системы). И именно с этой целью в нашей с вами программе мы должны выделить блок, в котором будут храниться эти команды. void setup(), а верней пространство внутри фигурных скобок этой функции, и является таким местом внутри Arduino скетча.

void setup()

{

Serial.begin(9600);// Инициирует последовательное соединение и задает скорость передачи данных в бит/c (бод)

irrecv.enableIRIn();// запуск ИК приемника

pinMode(pinI1,OUTPUT);//определяем пин А1 как интерфейс вывода

pinMode(pinI2,OUTPUT); //определяем пин А0 как интерфейс вывода

pinMode(speedpin1,OUTPUT); //определяем пин 6 как интерфейс вывода

pinMode(pinI3,OUTPUT);// //определяем пин А2 как интерфейс вывода

pinMode(pinI4,OUTPUT); //определяем пин А3 как интерфейс вывода

pinMode(speedpin2,OUTPUT); //определяем пин 5 как интерфейс вывода

pinMode(receiverpin,INPUT); //определяем пин 11 как интерфейс ввода

irrecv.enableIRIn();// запуск ИК приемник

}

// После вызова функции setup(), которая инициализирует и устанавливает первоначальные значения, функция loop() делает точь-в-точь то, что означает её название, и крутится в цикле, позволяя нашей программе совершать вычисления и реагировать на них. Используем её для активного управления платой Arduino.

void loop()

{

if (irrecv.decode(&results)) // если зажата клавиша

{

Serial.println(results.value, HEX); // выводим сигнал полученный с порта при нажатии кнопки в консоль в шестнадцатеричном значении

switch(results.value) // применяем оператор switch к значению считанному с ИК приемника

{

case ADVAN://если зажата клавиша «вперед»

goForward(1000); //движение вперед со скорость 1000 единиц

break; //завершаем выполнение кода и передаем управление следующему оператору после switch

case BAC:// если зажата клавиша «назад»

goBackward(1000); // движение назад со скорость 1000 единиц

break; //завершаем выполнение кода и передаем управление следующему оператору после switch

case left:// если зажата клавиша «влево»

goLeft(1000); // крутимся влево со скорость 1000 единиц

break; //завершаем выполнение кода и передаем управление следующему оператору после switch

case right:// если зажата клавиша «вправо»

goRight(1000); //// крутимся вправо со скорость 1000 единиц

break; //завершаем выполнение кода и передаем управление следующему оператору после switch

case STO:// если зажата клавиша «stop»

stopa(); //остановка

break; //завершаем выполнение кода и передаем управление следующему оператору после switch

default: // если значение не совпало с кейсами

delay(100); // делаем задержку в 100 мс

Serial.println(results.value); //выводим значение с ИК приемника в консоль

}

irrecv.resume(); // Считываем следующее значение с ИК приемника

}

delay(100); //задержка в 100 мс (после оператора switch )

}

// Далее идут пользовательские функции, выполняются один раз при каждом вызове из блоков setup или loop

void goForward(int a)// функция движения вперед

{

analogWrite(speedpin1,a);// задать скорость ввода аналогового значения

analogWrite(speedpin2,a);// задать скорость ввода аналогового значения

digitalWrite(pinI1,LOW);// I1 выходной низкий уровень

digitalWrite(pinI2,HIGH);// I2 выходной высокий уровень

digitalWrite(pinI3,LOW);// I3 выходной низкий уровень

digitalWrite(pinI4,HIGH);// I4 выходной высокий уровень

delay(100);//задержка в 100мс

}

void goBackward(int b)// функция движения назад

{

analogWrite(speedpin1,b); // задать скорость ввода аналогового значения

analogWrite(speedpin2,b); // задать скорость ввода аналогового значения

digitalWrite(pinI1,HIGH); // I1 выходной высокий уровень

digitalWrite(pinI2,LOW); // I2 выходной низкий уровень

digitalWrite(pinI3,HIGH); // I3 выходной высокий уровень

digitalWrite(pinI4,LOW); // I4 выходной низкий уровень

delay(100); //задержка в 100мс

}

void goLeft(int c)//функция движения влево

{

analogWrite(speedpin1,c); // задать скорость ввода аналогового значения

analogWrite(speedpin2,c); // задать скорость ввода аналогового значения

digitalWrite(pinI1,LOW); // I1 выходной низкий уровень

digitalWrite(pinI2,HIGH); // I2 выходной высокий уровень

digitalWrite(pinI3,HIGH); // I3 выходной низкий уровень

digitalWrite(pinI4,LOW); // I4 выходной высокий уровень

delay(100); //задержка в 100мс

}

void goRight(int d)// функция движения вправо

{

analogWrite(speedpin1,d); // задать скорость ввода аналогового значения

analogWrite(speedpin2,d); // задать скорость ввода аналогового значения

digitalWrite(pinI1,HIGH); // I1 выходной высокий уровень

digitalWrite(pinI2,LOW); // I2 выходной низкий уровень

digitalWrite(pinI3,LOW); // I3 выходной высокий уровень

digitalWrite(pinI4,HIGH); // I4 выходной низкий уровень

delay(100); //задержка в 100мс

}

void stopa()// функция остановки

{

digitalWrite(pinI1,LOW); // I1 выходной низкий уровень (0 В на выходе)

digitalWrite(pinI2,LOW); // I2 выходной низкий уровень (0 В на выходе)

digitalWrite(pinI3,LOW); // I2 выходной низкий уровень (0 В на выходе)

digitalWrite(pinI4,LOW); // I2 выходной низкий уровень (0 В на выходе)

delay(100); //задержка в 100мс

}