МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

ОТЧЁТ

По лабораторной работе №1

по дисциплине «Аппаратное и программное обеспечение   
роботизированных систем»

ПРОВЕРИЛ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гай В.Е.

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Полякова И.В.

17-В-1

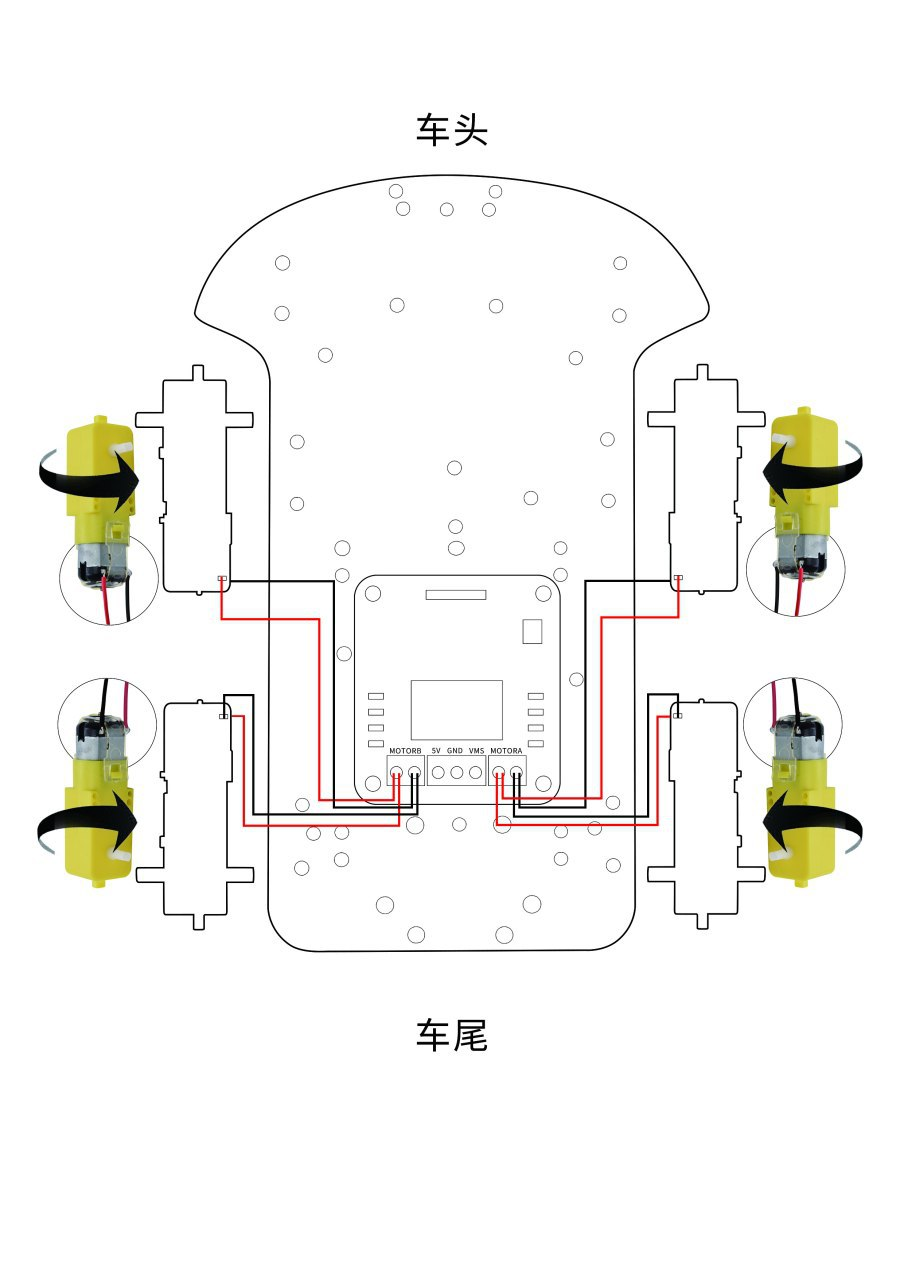
Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

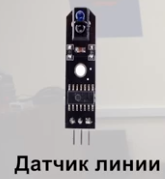
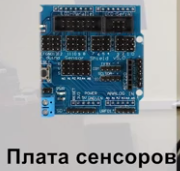
С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

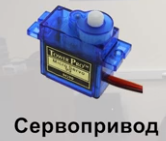
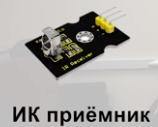
Нижний Новгород

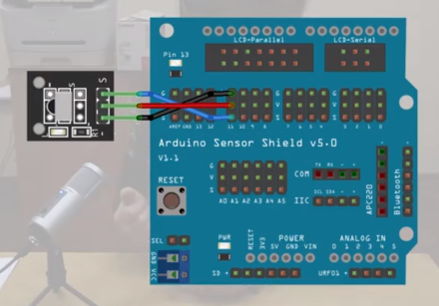
2020

**Часть 1. Построение робота.**



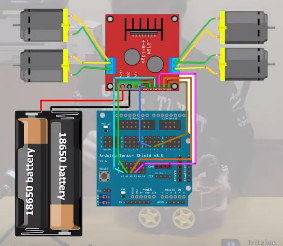
 

**Часть 2.Подключение платы Arduino UNO R3 к инфракрасному приёмнику (KY-022)** 

Он имеет 3 выхода(входа) желтый провод-земля, подключаем его на 11 позицию. G-ground, S-сигнал, V-питание. Соответствующие подключаем.

**Часть 3.Подключение драйвера двигателя (L298N) через расширительную плату (Sensor Shield 5.0).**

У драйевра двигателя есть na,nb выводы. Подключаем их к 6 и 5 каналу линии S. Есть выходы in2-а1(s),in3-a3,in4-a2. Выполнено подключение.



**Часть 4. Программа для робота.**

**Функции arduino void loop () и void setup ()**

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.

Чаще всего внутри void  setup указываются следующие инструкции:

* pinMode с указанием номера и типа пина. Это инструкция определяет  режим работы пинов ардуино.
* Serial.begin с указанием скорости (чаще всего, 9600). Эта инструкция инициализирует работы с последовательным портом на указанной скорости.
* Инструкции подключения и инициализации различных объектов библиотек ардуино. Например, servo.atach(6) укажет библиотеке, что сервопривод мы подключили к 9 пину, и все последующие действия код библиотеки будет совершать именно с этим портом.
* Инициализацию глобальных переменных, если мы по каким-то причинам не можем сделать это при определении переменных в глобальной области видимости.
* Выполнение других настроек и начальных значений для переменных и объектов.
* Функция loop это то место, куда мы должны поместить команды, которые  будут выполняться все время, пока включена плата Arduino. Начав выполнение с первой команды, микроконтроллер дойдет до конца и сразу же перепрыгнет в начало, чтобы повторить ту же последовательность. И так бесконечное число раз (до тех пор, пока на плату будет подан электричество). Наиболее уместный перевод английского слова loop в даном случае – это цикл (петля).
* По своей сути, void loop – это главная функция, точка входа в нашу программу. Arduino повторяет вызов этой функции миллионы раз в секунду. С помощью функций arduino void loop и void setup мы передаем микроконтроллеру  инструкции нашего скетча. Все то, что содержится внутри блока setup выполнится один раз. Содержимое блока loop будет выполняться в цикле до тех пор, пока останется включенным Arduino-контроллер.

## **Как запускается программа на Arduino**

Когда мы подключаем плату Arduino к питанию, то внутри него начинается весьма бурная деятельность встроенных микропрограмм. Микроконтроллер сконфигурирован так, что при запуске системы управление получает программа-загрузчик. Первое, что делает загрузчик – проверяет в течение 1-2 секунд, не начнется ли от пользователя отправка новой программы. Если процесс перепрограммирования начат, то скетч загружается в память и управление отдается ему. Если новых программ нет, то загрузчик выполняет ранее сохраненную программу.

Начав выполнение программы, Arduino выполняет ряд рутинных операций по инициализации и настройке среды окружения и только затем приступает к выполнению того самого кода, который  содержится в наших с вами скетчах. Таким образом, ардуино избавляет нас от необходимости помнить все детали архитектуры микропроцессора и сконцентрироваться на стоящей перед нами задаче.

Функция main() – это настоящая точка входа в программу, именно она вызывается первой. Как мы видим, в ней вызываются методы инициализации параметров и среды окружения, а затем и наши void setup() и, уже в цикле – void loop();

По идее, мы можем реализовать другую логику запуска и сделать внутри основного метода несколько функций loop, реализуя многопоточность.

#include <IRremote.h> // подключаем библиотеку для работы с ИК приемником

int receiverpin = 11;//Define the infrared receiving signal pin

int pinI1=A1;//Define I1 interface

int pinI2=A0;//Define I2 interface

int speedpin1=6;//The definition of EA (PWM speed control) interface

int pinI3=A3;//Define the I3 interface

int pinI4=A2;//Define the I4 interface

int speedpin2=5;//The definition of EB (PWM speed control) interface

#define ADVAN 0xFF629D //The remote control >> key 7 forward

#define BAC 0xFFA857 //The remote control << key 8 back

#define STO 0xFF02FD //The remote control > | key 6 stop

#define left 0xFF22DD //The remote control - key 5

#define right 0xFFC23D //The remote control + key 4

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

// Код в блоке setup выполняется один раз при каждом запуске микроконтроллера

void setup()

{

Serial.begin(9600);// Задаем скорость передачи данных в 9600 бод

irrecv.enableIRIn();// запускаем ИК приемник

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();// запускаем ИК приемник

}

// Код в блоке loop выполняется “по кругу” на всём протяжении работы микроконтроллера, начиная с момента завершения выполнения setup.

void loop()

{

if (irrecv.decode(&results)) // Если кнопка была нажата

{

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://проверка кнопки «вперед»

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

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

case BAC:// проверка кнопки «назад»

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

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

case left:// проверка кнопки «влево»

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

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

case right:// проверка кнопки «вправо»

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

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

case STO:// проверка кнопки «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 езды вперед

{

analogWrite(speedpin1,a);//подаем на аналоговый порт «6» значение «скорости»

analogWrite(speedpin2,a);//подаем на аналоговый порт «5» значение «скорости»

digitalWrite(pinI1,LOW);// 0 В на выходе

digitalWrite(pinI2,HIGH);// 5 В на выходе

digitalWrite(pinI3,LOW);// 0 В на выходе

digitalWrite(pinI4,HIGH);// 5 В на выходе

delay(100);//задержка в 100мс

}

void goBackward(int b)// функция езды назад

{

analogWrite(speedpin1,b); //подаем на аналоговый порт «6» значение «скорости»

analogWrite(speedpin2,b); //подаем на аналоговый порт «5» значение «скорости»

digitalWrite(pinI1,HIGH); // 0 В на выходе

digitalWrite(pinI2,LOW); // 0 В на выходе

digitalWrite(pinI3,HIGH); // 0 В на выходе

digitalWrite(pinI4,LOW); // 0 В на выходе

delay(100); //задержка в 100мс

}

void goLeft(int c)//функция кручения «влево»

{

analogWrite(speedpin1,c); //подаем на аналоговый порт «6» значение «скорости»

analogWrite(speedpin2,c); //подаем на аналоговый порт «5» значение «скорости»

digitalWrite(pinI1,LOW); // 0 В на выходе

digitalWrite(pinI2,HIGH); // 0 В на выходе

digitalWrite(pinI3,HIGH); // 0 В на выходе

digitalWrite(pinI4,LOW); // 0 В на выходе

delay(100); //задержка в 100мс

}

void goRight(int d)// фцнкция кручения вправо

{

analogWrite(speedpin1,d); //подаем на аналоговый порт «6» значение «скорости»

analogWrite(speedpin2,d); //подаем на аналоговый порт «5» значение «скорости»

digitalWrite(pinI1,HIGH); // 0 В на выходе

digitalWrite(pinI2,LOW); // 0 В на выходе

digitalWrite(pinI3,LOW); // 0 В на выходе

digitalWrite(pinI4,HIGH); // 0 В на выходе

delay(100); //задержка в 100мс

}

void stopa()// функция остановки

{

digitalWrite(pinI1,LOW); // 0 В на выходе

digitalWrite(pinI2,LOW); // 0 В на выходе

digitalWrite(pinI3,LOW); // 0 В на выходе

digitalWrite(pinI4,LOW); // 0 В на выходе

delay(100); //задержка в 100мс

}

Simple\_control\_system - исходная программа для управления движением робота с ИК пульта PWM\_control\_system - управление скоростью движения робота с ИК пульта.

Программа для управления роботом с ИК пульта. Кнопка 7 - движение вперёд Кнопка 8 - движение назад Кнопка 6 - остановка робота Кнопка 5 - движение налево Кнопка 4 - движение направо