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 17-в-1: Максимова О. В.

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

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

Нижний Новгород

2020

**Arduino?**

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте [языка программирования](http://arduino.ru/Reference), а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

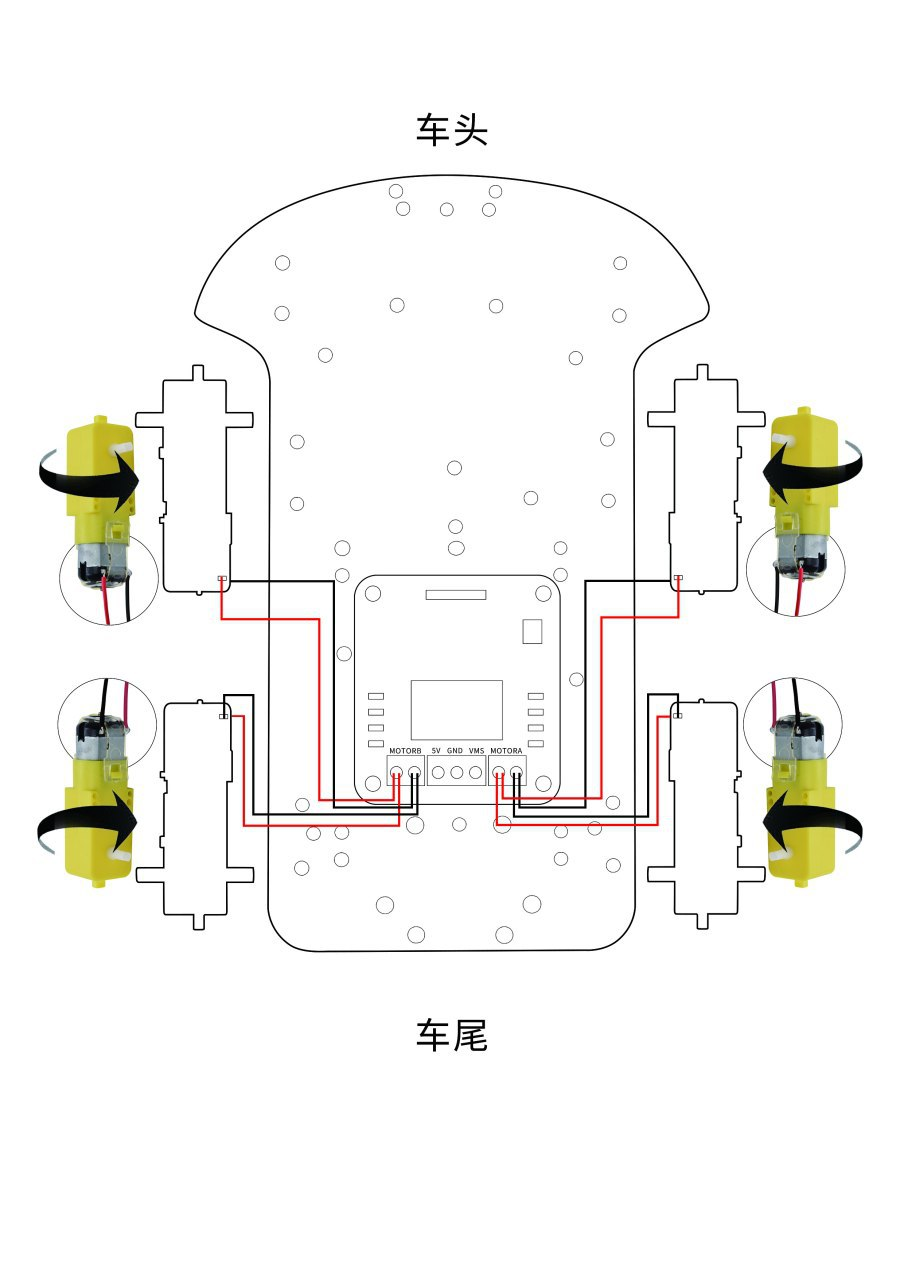
Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе  Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

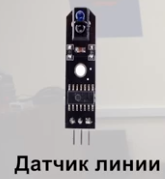
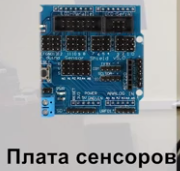
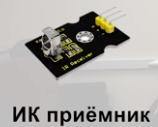
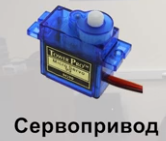
Микроконтроллер на плате программируется при помощи [языка Arduino](http://arduino.ru/Reference) (основан на языке [Wiring](http://out.arduino.ru/?redirect=http%3A%2F%2Fwiring.org.co%2F&baseU=http%3A%2F%2Farduino.ru%2F" \o "Wiring среда разработки" \t "_blank)) и среды разработки Arduino (основана на среде [Processing](http://out.arduino.ru/?redirect=http%3A%2F%2Fwww.processing.org%2F&baseU=http%3A%2F%2Farduino.ru%2F" \t "_blank)). Проекты устройств, основанные на Arduino, могут работать самостоятельно, либо же взаимодействовать с программным обеспечением на компьютере (напр.: Flash, Processing, MaxMSP). Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Программное обеспечение доступно для [бесплатного скачивания](http://out.arduino.ru/?redirect=http%3A%2F%2Farduino.cc%2Fen%2FMain%2FSoftware&baseU=http%3A%2F%2Farduino.ru%2F). Исходные чертежи схем (файлы CAD) являются общедоступными, пользователи могут применять их по своему усмотрению.

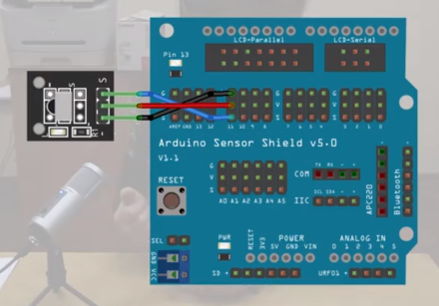
В 2006  Arduino получила признание в категории Digital Communities на фестивале Ars Electronica Prix .

[](http://arduino.ru/sites/default/files/Hardware/updated/uno_front.jpg)

**Построение робота.**



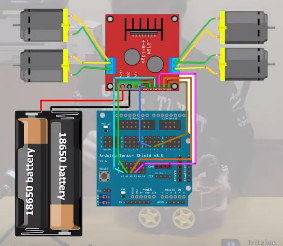
     

**Подключение платы Arduino UNO R3 к инфракрасному приёмнику (KY-022)** 

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

**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.



**Программа для робота.**

Функции arduino void loop () и void setup ()

Функции void loop () и void setup () в Aduino IDE — это первое с чем сталкивается любой, кто начинает знакомство с языком программирования микроконтроллеров Ардуино. Разберем для чего нужны в скетче данные функции, что следует размещать внутри циклов. Данные правила относятся ко всем платам — от [Arduino UNO](https://xn--18-6kcdusowgbt1a4b.xn--p1ai/%d0%b0%d1%80%d0%b4%d1%83%d0%b8%d0%bd%d0%be-%d1%83%d0%bd%d0%be/" \o "Arduino UNO) до Arduino MEGA и перед изучением языка Aduino IDE, следует подробно изучить данные функции.

Процедуры void loop и void setup

Данные циклы должны быть в каждом скетче и вызываться только один раз, даже если один из циклов не используется. Дело в том, что при запуске микроконтроллера Arduino, начинают работать встроенные микропрограммы, которые первым делом проверяют не началась загрузка новой программы с компьютера. Если пользователь не начал прошивку, то контроллер начинает выполнять ранее загруженный скетч.

Оба цикла вызываются встроенной функцией main() из файла main.cpp. При этом функция void setup () вызывается один раз, а и void loop() вызывается в цикле for бесконечное количество раз. Если в вашем скетче будет присутствовать более одного раза функция void setup() или void loop(), то при [компиляции Aduino IDE](https://xn--18-6kcdusowgbt1a4b.xn--p1ai/%d0%be%d1%88%d0%b8%d0%b1%d0%ba%d0%b8-%d0%bf%d1%80%d0%b8-%d0%ba%d0%be%d0%bc%d0%bf%d0%b8%d0%bb%d1%8f%d1%86%d0%b8%d0%b8-%d0%b0%d1%80%d0%b4%d1%83%d0%b8%d0%bd%d0%be/) выдаст ошибку: redefinition of ‘void setup()‘ или redefinition of ‘void loop()’, соответственно.

Описание void setup () в Arduino

Функция setup в Aduino IDE имеет следующую конструкцию:

void setup() {

// код должен располагаться между фигурных скобок

}

Фигурные скобки указывают, где начало и конец цикла, поэтому все команды должны располагаться между ними. Если вы случайно удалите или поставите лишнюю фигурную скобку, то при компиляции скетча будет выходить ошибка. Процедура void setup вызывается один раз и ее используют для назначения режима работы пинов или команд, которые необходимо выполнить только в момент загрузки программы.

Описание void loop () в Arduino

После выполнения цикла setup, программа переходит в цикл loop, который будет повторяться до тех пор, пока на плату подано питание. Если цикл содержит одну команду, то она будет выполняться тысячи раз в секунду. Если вы решите написать скетч для [мигания светодиодом](https://xn--18-6kcdusowgbt1a4b.xn--p1ai/%d0%bc%d0%b8%d0%b3%d0%b0%d0%bd%d0%b8%d0%b5-%d1%81%d0%b2%d0%b5%d1%82%d0%be%d0%b4%d0%b8%d0%be%d0%b4%d0%be%d0%bc/) на Arduino, то необходимо добавлять в код задержку для выполнения программы, иначе мигания светодиода не будет заметно.

Функция loop в Aduino IDE имеет следующую конструкцию:

void loop() {

// основной код программы располагается здесь

}

Таким образом, если вам необходимо при запуске программы включить один раз светодиод для индикации работы устройства на микроконтроллере Arduino Nano, то команду лучше написать в цикле void setup(). Если в программе необходимо выполнять какое-то действие постоянно, например, получать данные с ультразвукового дальномера [HC-SR04](https://xn--18-6kcdusowgbt1a4b.xn--p1ai/%d1%83%d0%bb%d1%8c%d1%82%d1%80%d0%b0%d0%b7%d0%b2%d1%83%d0%ba%d0%be%d0%b2%d0%be%d0%b9-%d0%b4%d0%b0%d1%82%d1%87%d0%b8%d0%ba-%d0%b0%d1%80%d0%b4%d1%83%d0%b8%d0%bd%d0%be/), то команду следует располагать в цикле void loop ().

## **Как запускается программа на Arduino**

Когда мы подключаем плату Arduino к питанию, то внутри него начинается 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 робота

int receiverpin = 11;// вывод ИК – приемника к цифровому входу 11 на плате Arduino

int pinI1=A1;//кнопка на входе A1

int pinI2=A0;// кнопка на входе A0

int speedpin1=6;// цифровые входы для управления скоростью

int pinI3=A3;// кнопка на входе A3

int pinI4=A2;// кнопка на входе A2

int speedpin2=5;//цифровые входы для управления скоростью

#define ADVAN 0xFF629D // даем имя константе >> key 7 вперелд

#define BAC 0xFFA857 // даем имя константе << key 8 назад

#define STO 0xFF02FD // даем имя константе > | key 6 остановка

#define left 0xFF22DD // даем имя константе- key 5

#define right 0xFFC23D // даем имя константе + 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)// едем вперед

{

analogWrite(speedpin1,a);// Выдает аналоговую величину ([ШИМ волну](http://arduino.ru/Tutorial/PWM)) на порт вход/выхода «6» значение «скорости»

analogWrite(speedpin2,a);// Выдает аналоговую величину ([ШИМ волну](http://arduino.ru/Tutorial/PWM)) на порт вход/выхода «5» значение «скорости»

digitalWrite(pinI1,LOW);// Подает HIGH или LOW значение на цифровой вход/выход (pin) 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)// едем назад

{

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

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

digitalWrite(pinI1,HIGH); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 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); // Выдает аналоговую величину ([ШИМ волну](http://arduino.ru/Tutorial/PWM)) на порт вход/выхода «6» значение «скорости»

analogWrite(speedpin2,c); // Выдает аналоговую величину ([ШИМ волну](http://arduino.ru/Tutorial/PWM)) на порт вход/выхода «5» значение «скорости»

digitalWrite(pinI1,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI2,HIGH); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI3,HIGH); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI4,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

delay(100); //задержка в 100мс

}

void goRight(int d)// фцнкция кручения вправо

{

analogWrite(speedpin1,d); || Выдает аналоговую величину ([ШИМ волну](http://arduino.ru/Tutorial/PWM)) на порт вход/выхода «6» значение «скорости»analogWrite(speedpin2,d); // Выдает аналоговую величину ([ШИМ волну](http://arduino.ru/Tutorial/PWM)) на порт вход/выхода «5» значение «скорости»

digitalWrite(pinI1,HIGH); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI2,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI3,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI4,HIGH); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

delay(100); //задержка в 100мс

}

void stopa()// функция остановки

{

digitalWrite(pinI1,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI2,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI3,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 0 В на выходе

digitalWrite(pinI4,LOW); // Подает HIGH или LOW значение на цифровой вход/выход (pin) 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