

Научно-исследовательская работа

Технологии

ЛЕГКИЙ РОБОТ ЭКСПЕРИМЕНТАТОР KILLBOT-0 AKAME

Выполнил:

Комиссаров Владимир Александрович

Учащийся группы 2МРО2

МАДК им. Николаева, Россия, г. Москва

МАДК им. Николаева, Россия, г. Москва

В этой работе я хочу представить свой лучший проект, над которым я работаю уже почти год. Здесь я представляю результаты, которые мне удалось достичь. Так же своим проектом я хочу показать то что в наше время робототехника и программируемая электроника довольно доступная и при желании каждый сможет построить своего робота.

Модель: Killbot-0.

Имя: Акаме.

Класс: лёгкий экспериментатор.

Назначение: разработка и отработка электронной базы для будущих моделей Killbot.

Краткое описание:

Легкий вездеходный робот на упрощённой марсоходной базе способный выполнять простые задачи в труднодоступных или опасных местах.

Подвеска и корпус:

Изготовлены из вспененного ПВХ пластика, деталей распечатанных на 3D принтере и алюминиевых трубок. Элементы подвески скреплены болтами и гайками М3. Каждое колесо имеет свой мотор-редуктор.



(рис 1) Killbot-0 Акаме

длинна	ширина	Высота (без антенны)	Дорожный просвет	Высота корпуса	Ширина корпуса	длинна корпуса	Диаметр колёс
370мм	360мм	171мм	121мм	50мм	134мм	235мм	60мм

Модификации:

0.0: привод 4 на 6, диаметр колёс: 42мм.

0.1: привод 6 на 6.

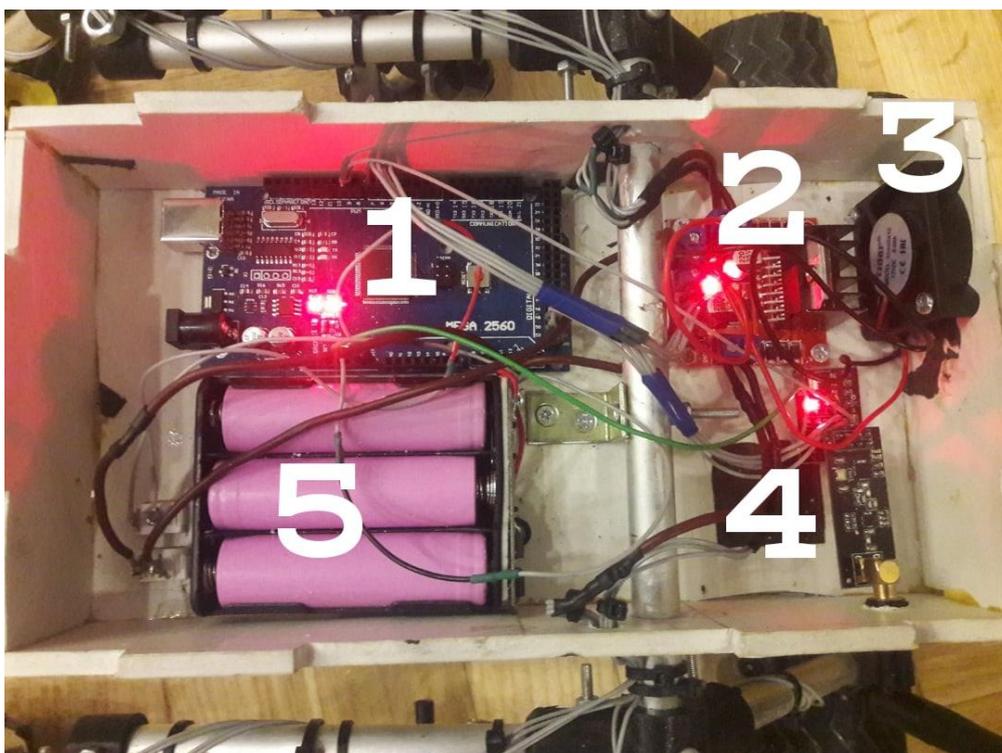
0.2: диаметр колёс: 60мм

0.3: улучшено крепление колёс.

Электроника:

Построена на основе Arduino Mega (пульт на Arduino NANO).

Компоненты:



(рис 2) компоненты системы управления.

1	2	3	4	5
Arduino Mega	Драйвер L298N	Куллер для охлаждения драйвера	Модуль радиообмена Nrf24L01	Сборка аккумуляторов 18650 с платой защиты. 12V.

Модификации:

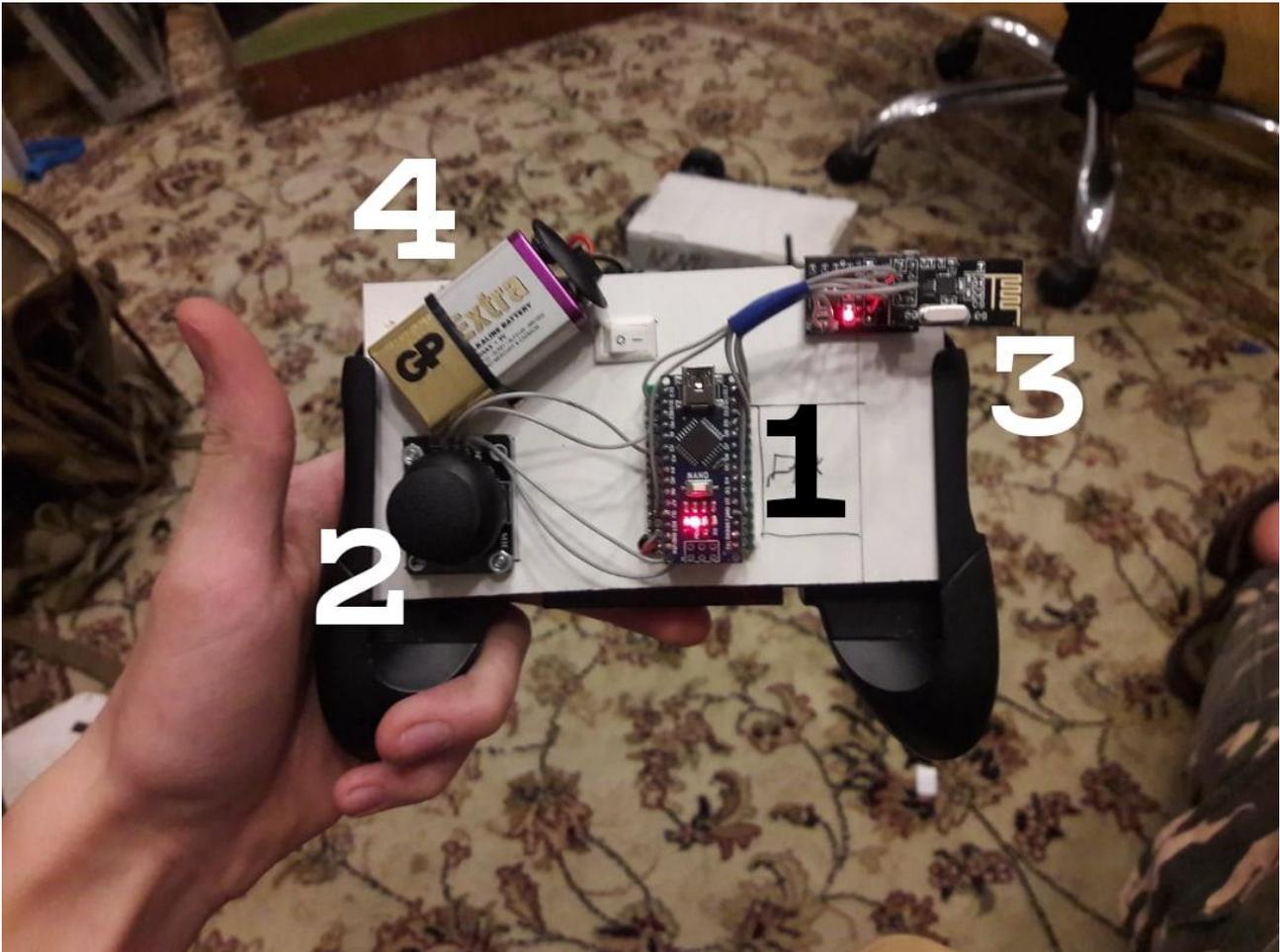
0.0: контроллер Arduino NANO, драйвер: самодельный шаговый переключатель. Вместо модуля nRF24L01 установлен ультразвуковой сенсор дистанции. Бортовое напряжение: 5V.

0.1: драйвер MX1508, установлен модуль радиообмена, бортовое напряжение 5V и 9V.

0.2: драйвер L298N, бортовое напряжение 12V и 5V.

1.0: контроллер: Arduino Mega, добавлен куллер охлаждения драйвера.

Пульт:



(рис 3) пульт

1	2	3	4
Arduino NANO	Джойстик	Модуль радиообмена Nrf24l01(без антенны)	Источник питания 9V

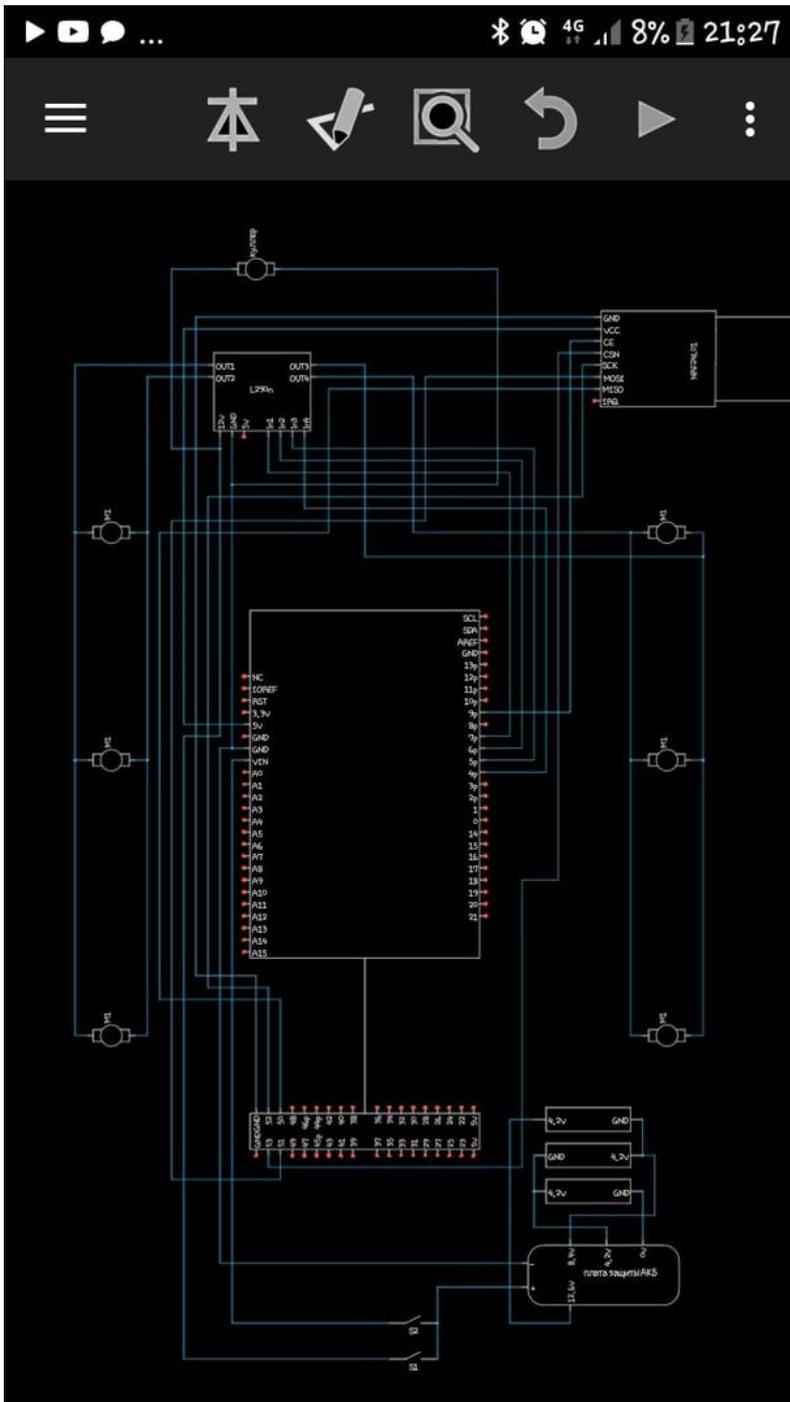
Модификации:

0.0: контроллер Arduino NANO, кнопочная крестовина для взаимодействия, модуль радиообмена nRF24L01, напряжение 9V.

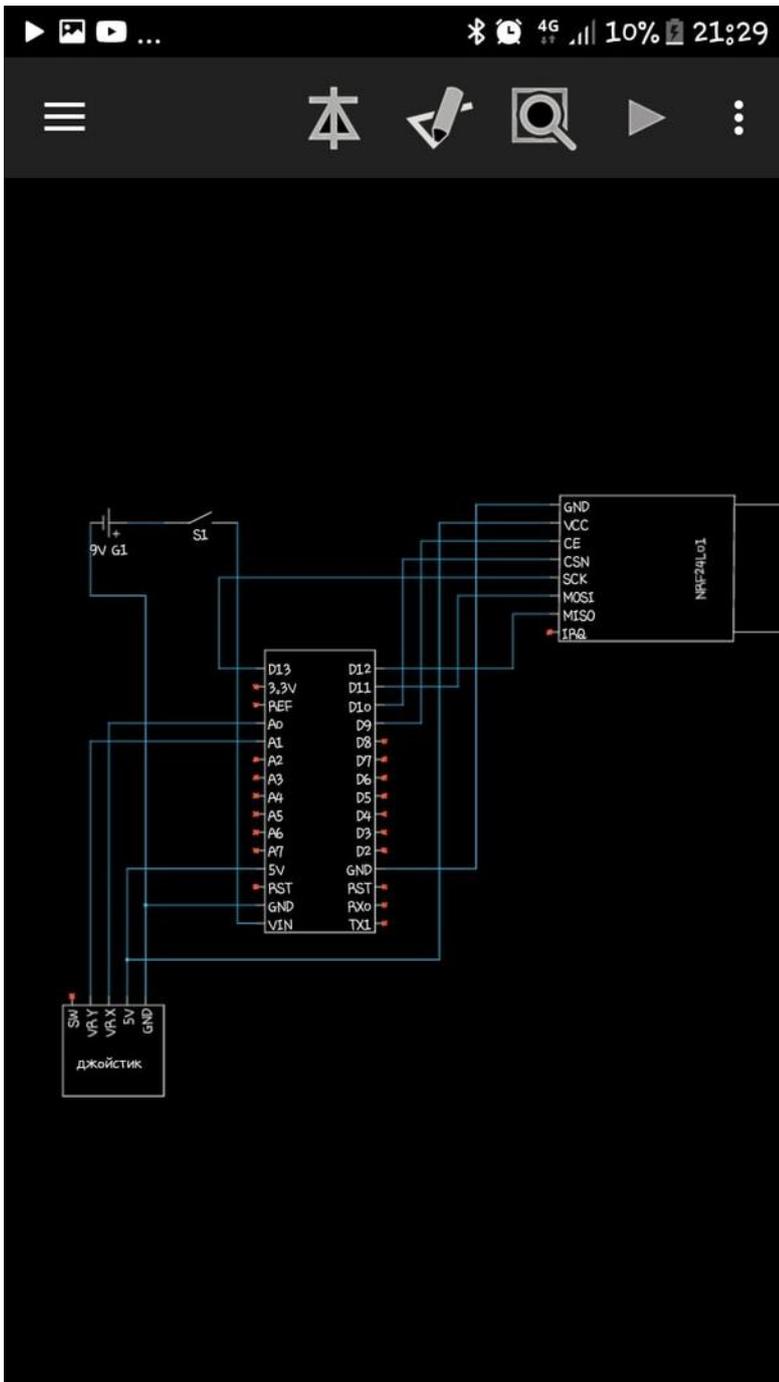
1.0: установлен джойстик, модуль радиообмена nRF24L01 без антенны, обновление корпуса.

Схемы подключения:

Схемы построены в программе Quick Copper.



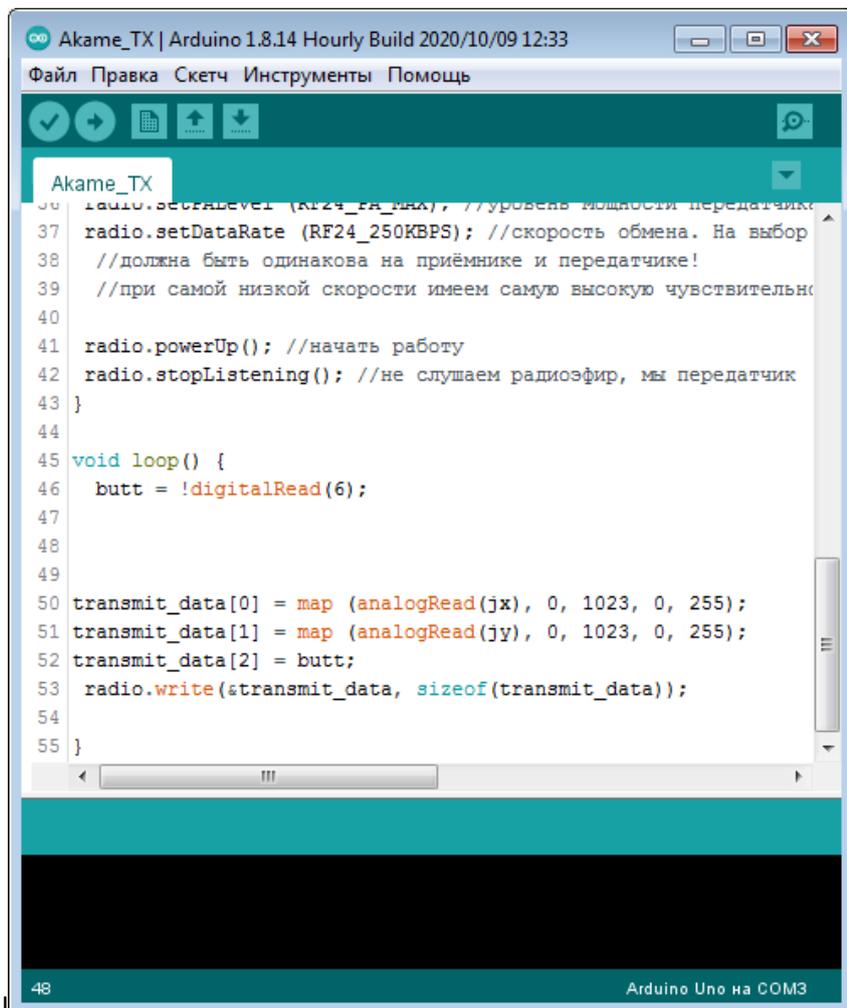
(рис 4) схема системы управления 1.0



(рис 5) схема пульта 1.0

Скетчи:

Скетчи написаны на основе двух скетчей из примера скетча AlexGyver и примера библиотеки GyverMotor. Я показываю только ту часть скетча, которая отвечает за управление роботом. Остальное это настройки библиотек и радиопередачи.



```
36 radio.setPowerLevel (RF24_PA_MAX); //уровень мощности передатчика
37 radio.setDataRate (RF24_250KBPS); //скорость обмена. На выбор
38 //должна быть одинакова на приёмнике и передатчике!
39 //при самой низкой скорости имеем самую высокую чувствительность
40
41 radio.powerUp(); //начать работу
42 radio.stopListening(); //не слушаем радиоэфир, мы передатчик
43 }
44
45 void loop() {
46   butt = !digitalRead(6);
47
48
49
50   transmit_data[0] = map (analogRead(jx), 0, 1023, 0, 255);
51   transmit_data[1] = map (analogRead(jy), 0, 1023, 0, 255);
52   transmit_data[2] = butt;
53   radio.write(&transmit_data, sizeof(transmit_data));
54
55 }
```

(рис б) часть скетча отправки данных с джойстика.

(Читаю с момента void loop())

46. обозначаю кнопку на 6 цифровом пине.

50. массиву данных 0 присваиваю значения оси X с джойстика и преобразую максимальное значение из 1023 в 255.

51. Тоже самое что в 50 но по оси джойстика Y.

52. массиву данных 2 присваиваю значение кнопки.

53. отправить массивы данных по радио.

```
54  
55  
56 void loop() {  
57   digitalWrite(0, 1);  
58   byte pipeNo;  
59   while( radio.available(&pipeNo)){ // слушаем эфир со все  
60     radio.read( &recieved_data, sizeof(recieved_data) ); // т  
61  
62     X = map(recieved_data[0], 0, 255, 0,1023);  
63     Y = map(recieved_data[1], 0, 255, 0,1023);  
64  
65     int signalX = 255-X / 2;  
66     int signalY = 255-Y / 2;  
67  
68     int dutyR = signalY + signalX;  
69     int dutyL = signalY - signalX;  
70  
71     motorR.setSpeed(dutyR);  
72     motorL.setSpeed(dutyL);  
73  
74  
75     digitalWrite(14, recieved_data[2]);  
76  
77  
78  
79  
80
```

(рис 7) часть скетча приёма данных и преобразования по танковой схеме.

(Читаю с момента void loop())

57. пин 0 равен 1(на этом пине будет 5V)

58 – 60. Принять массивы данных.

62. принимаю и преобразую обратно значения оси X с джойстика.

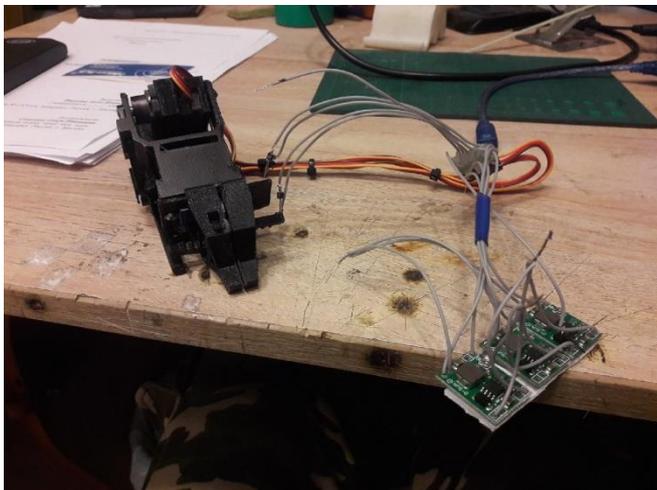
63. Тоже самое, но ось джойстика Y.

62 – 72. Преобразование значений джойстика в сигналы для драйвера по танковой схеме.

75. пин 14 равен значению кнопки.

Дополнительное оборудование:

На данный момент находится в разработке. В планах установка небольшого пятиступенчатого манипулятора и FPV камеры на двухосевом подвесе.



(рис 8) манипулятор

Заключение:

Благодаря работе над этим роботом мне удалось неплохо освоить Arduino и несколько модулей для неё. В будущем полученный опыт я использую для постройки уже полноценных роботов способных работать на большом расстоянии, иметь продолжительный срок автономной работы и способный выполнять сложные задачи. Модель Killbot-0 является моим первым проектом относящихся к робототехнике и работу над ним я начал, имея минимальный опыт в работе с Arduino. Но получая информацию из интернета (в основном с канала AlexGyver) мне удалось достичь этого результата. Проект Killbot-0 ещё не завершён и имеет множество недочётов, и моя цель методом проб и ошибок исправить их.

Источники:

1. <https://alexgyver.ru/>
2. https://www.youtube.com/channel/UCgtAOyEQdAyjvm9ATCi_Aig
3. <http://wiki.amperka.ru/продукты:arduino-mega-2560>
4. https://www.youtube.com/channel/UCRKRGo0367_uweBIZ8PF4Nw