

УДК 519.24

## РАДИОУПРАВЛЯЕМАЯ МОДЕЛЬ НА БАЗЕ МИКРОКОНТРОЛЛЕРА ARDUINO MEGA 2560\*

А.И. НЕЗВАНОВ<sup>1</sup>, И.И. МОСКВИЧЕВ<sup>2</sup>, Е.Д. ЖАРОВ<sup>3</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент факультета автоматики и вычислительной техники. E-mail: nezvanov@mail.ru

<sup>2</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент факультета автоматики и вычислительной техники. E-mail: ivanmoskvichev96@mail.ru

<sup>3</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент факультета автоматики и вычислительной техники. E-mail: edar.vih.esrr@yandex.ru

На создание данного проекта нас воодушевили прошедшие соревнования по робототехнике в НГТУ. Небольшие, но умные роботы могли шагать, бороться, ездить и перевозить грузы по установленным трассам и препятствиям. Главной задумкой было создать модель машины-робота, управляемую микроконтроллером, запрограммированным нами самостоятельно, так как 90 % роботов, представленных на соревнованиях, были созданы из наборов Lego Mindstorms. Мы посчитали, что будет намного интересней сделать уникальную машину, полностью контролируемую своим алгоритмом. На данный момент радиопередача и беспроводные технологии прочно вошли в нашу жизнь. Многие, несмотря на активный прогресс и век большой информации, не имеют даже малейшего представления о том, как работают эти системы и технологии. Одной из целей в данной работе является научиться применять беспроводные технологии для решения прикладных задач и собрать модель для практического применения полученных навыков. Управление радиоуправляемой моделью движущегося заднеприводного автомобиля реализовано с помощью микроконтроллера Arduino Mega 2560, а непосредственная передача команд управления – с помощью микроконтроллера Arduino Uno R3 и джойстика, являющегося, по сути, двумя потенциометрами, соединенными в одной ручке. Для питания модели используется аккумулятор на 12 вольт для двигателей, для низковольтных цепей (микроконтроллер) используется полупроводниковый стабилизатор напряжения на 5 вольт, а для питания контроллера управления используется USB-разъем от компьютера. Мы искренне надеемся, что данный материал будет полезен для начинающих студентов, желающих изучить и применить данную технологию на своих радиоуправляемых устройствах. В будущем планируется развивать проект и использовать данную технологию в экспериментальных и научных целях. Данный материал предназначен исключительно в ознакомительных целях.

---

\* Статья получена 15 марта 2016 г.

**Ключевые слова:** микроконтроллер Arduino Mega 2560, робототехника, Motor shield, сервопривод, трансивер, Serial Parallel Interface (SPI), радиоуправление, программирование

DOI: 10.17212/2307-6879-2016-2-88-98

## ВВЕДЕНИЕ

В современной жизни беспроводные технологии прочно вошли в нашу жизнь, порой мы даже не задумываемся, используя их в повседневной жизни. Самым простым примером тому являются мобильные телефоны или беспроводные сети Wi-Fi. Целью данной работы было создать собственноручно платформу, имеющую возможность управления с помощью беспроводной радиосвязи и представляющую собой базу для дальнейшего использования в других проектах.

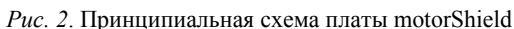
## 1. ПЛАТФОРМА ARDUINO MEGA 2560

Arduino Mega построена на микроконтроллере ATmega 2560. Плата имеет 54 цифровых входов/выходов (14 из которых могут использоваться как выходы ШИМ с разрядностью 8 бит), 16 аналоговых входов, 4 последовательных порта UART, кварцевый генератор на 16 МГц, USB-коннектор, разъем питания, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареей. Arduino Mega 2560 совместима со всеми платами расширения, разработанными для платформ Uno или Duemilanove.

## 2. КОМПЛЕКТУЮЩИЕ ДЛЯ СБОРКИ

- Arduino Mega 2560 (рис. 1).
- MotorShield (плата расширения для Arduino, управляющая двигателями постоянного тока) (рис. 2).
- Сервопривод для управления (TowerPro SG90).
- Трансивер nRF24L01 2.4GHz.
- Прочие радиодетали (резисторы, светодиоды-индикаторы, стабилизаторы напряжения).





В ходе сборки также был задействован еще один МК Arduino Uno, трансивер и джойстик (представляет собой два потенциометра) для создания передатчика управляющих сигналов на платформу. Дальность передачи (стабильной) сигнала была установлена опытным путем и составила на открытом пространстве около 50 метров, в закрытом пространстве 15 метров (сквозь три железобетонные стены). При потере сигнала платформа останавливается и ожидает возобновления управления.

- 1) 9 – сервопривод рулевого управления;
- 2) 4 – clockPin (SPI) – тактовый выход;
- 3) 8 – dataPin (SPI) – выход битовых импульсов;
- 4) 12 – latchPin (SPI) – выход-зашелка, переводящий сдвиговый регистр (74HC595N на плате motorShield) в состояние вывода или состояние записи;
- 5) 30, 31, 32 – светодиоды состояния;

- 6) 49 – CE (nRF24101);
- 7) 53 – CSN(nRF24101);
- 8) 52 – SCK(nRF24101);
- 9) 51 – MOSI (nRF24101);
- 10) 50 – MISO (nRF24101);

11) 3, 5, 6, 11 – выходы управления скоростью вращения моторов. Регулировка осуществляется с помощью ШИМ.

Для управления motorShield было решено написать свою функцию, так как существующие библиотеки весьма сложны для понимания и не имеют документацию на русском языке.

Функция **motorSpeed(short motor, short speed)** получает параметры и устанавливает выводы микроконтроллера в нужное состояние.

Функция **motorDirection(byte M1, byte M2, byte M3, byte M4)** получает для каждого из моторов байт: 0 – стоп, 1 – вперед, 2 – назад и через SPI устанавливает выводы платы motorShield в нужное состояние.

Готовая модель изображена на рис. 3.

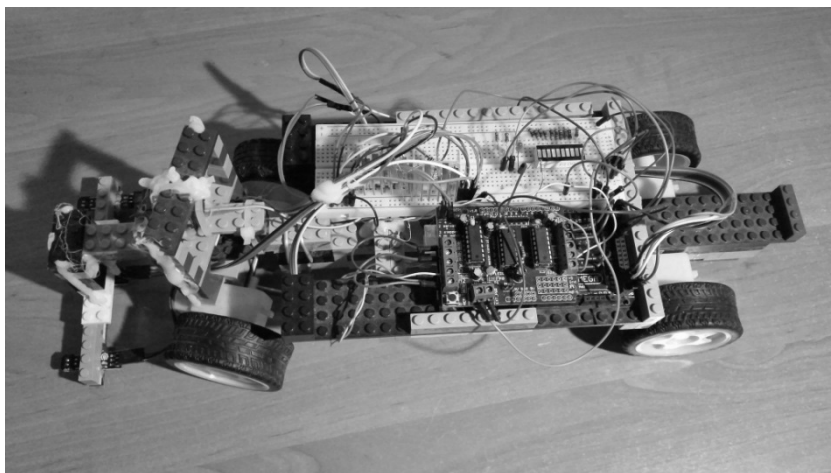


Рис. 3. Модель в сборе

## 4. ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <Servo.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
```

```
#define CE_PIN 49
#define CSN_PIN 53

const uint64_t pipe = 0xE8E8F0F0E1LL;
RF24 radio(CE_PIN, CSN_PIN);
unsigned short joystick[3];
int speed = 0;
/*Pins for SPI*/
int latchPin = 12;
int clockPin = 4;
int dataPin = 8;
int temp;
Servo steer;

void motorSpeed(short motor, short speed){
    short temp;
    if(speed == 0) temp = 0;
    else temp = map(speed,1,9,180,255);
    if(motor == 1) analogWrite(11,temp);
    if(motor == 2) analogWrite(3,temp);
    if(motor == 3) analogWrite(5,temp);
    if(motor == 4) analogWrite(6,temp);
}

void motorDirection(byte M1, byte M2, byte M3, byte M4){
    digitalWrite(latchPin, LOW);
    if(M4 == 1) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M3 == 1) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M4 == 2) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M2 == 1) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M1 == 1) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M1 == 2) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M2 == 2) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
    if(M3 == 2) digitalWrite(dataPin, HIGH);
    else digitalWrite(dataPin, LOW);
```

```
digitalWrite(clockPin, HIGH);digitalWrite(clockPin, LOW);
digitalWrite(latchPin, HIGH);
}

void setup()
{
  steer.attach(9);
  radio.begin();
  radio.openReadingPipe(1,pipe);
  radio.startListening();;
  pinMode(30, OUTPUT);
  pinMode(31, OUTPUT);
  pinMode(32, OUTPUT);
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(7,OUTPUT);
  digitalWrite(7,LOW);
  pinMode(11,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  motorSpeed(1,0);
  motorSpeed(2,0);
  motorSpeed(3,0);
  motorSpeed(4,0);
}

void loop()
{
  int temp = 0;
  if ( radio.available() )
  {
    bool done = false;
    while (!done)
    {
      done = radio.read( joystick, sizeof(joystick) );
      steer.write(map(joystick[2],0,1023,0,180));
      if(joystick[1] > 600 || joystick[1] < 400){
        digitalWrite(32,HIGH);
        digitalWrite(31,LOW);
        digitalWrite(30,LOW);
        if(joystick[1] > 600){
          temp = map(joystick[1],600,1000,0,9);
          motorSpeed(1,temp);
          motorSpeed(2,temp);
          motorSpeed(3,temp);
          motorSpeed(4,temp);
          motorDirection(1,1,1,1);
        }
      }
    }
  }
}
```

```
    } else if(joystick[1] < 400){
        temp = map(joystick[1],0,400,9,0);
        motorSpeed(1,temp);
        motorSpeed(2,temp);
        motorSpeed(3,temp);
        motorSpeed(4,temp);
        motorDirection(2,2,2,2);
    }
} else{
    temp = 0;
    motorDirection(0,0,0,0);
    digitalWrite(31,HIGH);
    digitalWrite(30,LOW);
    digitalWrite(32,LOW);
}
}
}
else
{
    motorDirection(0,0,0,0);
    digitalWrite(30,HIGH);
    digitalWrite(31,LOW);
    digitalWrite(32,LOW);
}
}
```

## ЗАКЛЮЧЕНИЕ

В данной работе нами были изучены принципы программирования микроконтроллера Arduino Mega 2560, а также проведена работа по интегрированию МК в собственные разработки, т. е. связь программы и физических действий устройства.

## СПИСОК ЛИТЕРАТУРЫ

1. *Banzy M.* Getting started with Arduino. – Sebastopol, CA: Maker Media, 2011. – 128 с.
2. *Brian W.E.* Arduino programming notebook. – San Francisco, CA: Creative Commons, 2007. – 35 p.
3. *McRoberts M.* Beginning Arduino. – New York, NY: Apress, 2010. – 459 p.
4. *Соммер У.* Программирование микроконтроллерных плат Arduino/Free-duino. – СПб.: БХВ-Петербург, 2012. – 256 с.
5. *Днищенко В.А.* Дистанционное управление моделями. – СПб.: Наука и техника, 2007. – 456 с.



6. *Boxall J.* Arduino workshop. – San Francisco: No starch press, 2013. – 394 p.
7. *Банци М.* Arduino для начинающих волшебников. – М.: Рид Групп, 2012. – 128 с.
8. *Иго Т.* Arduino, датчики и сети для связи устройств. – СПб.: БХВ-Петербург, 2015. – 544 с.
9. *Монк С.* Програмируем Arduino: основы работы со скетчами. – СПб.: Питер, 2016. – 176 с.
10. *Петин В.* Проекты с использованием контроллера Arduino. – СПб.: БВХ-Петербург, 2014. – 400 с.
11. *Margolis M.* Arduino cookbook. – Sebastopol, CA: O'Reilly, 2012. – 724 p.
12. *Петин В.* Проекты с использованием контроллера Arduino. – 2-ое изд. – СПб.: БВХ-Петербург, 2015. – 464 с.
13. *Blum J.* Exploring Arduino: tools and techniques for engineering wizardry. – Indianapolis: Wiley and Sons, 2013. – 357 p.
14. *Boxall J.* Arduino Workshop: a hands-on introduction with 65 projects. – Kindle ed. – San Francisco: No Strach Press, 2008. – 250 p.
15. *Savagard E.* Arduino: 101 beginners guide: how to get started with your Arduino (tips, tricks, projects and more!). – Kindle ed. – Sebastopol, CA: O'Reilly Media, 2008. – 38 p.

**Незванов Александр Игоревич**, студент факультета автоматики и вычислительной техники Новосибирского государственного технического университета. E-mail: nezvanov@mail.ru

**Москвичёв Иван Игоревич**, студент факультета автоматики и вычислительной техники Новосибирского государственного технического университета. E-mail: ivanmoskvichev96@mail.ru

**Жаров Евгений Дмитриевич**, студент факультета автоматики и вычислительной техники Новосибирского государственного технического университета. E-mail: edar.vih.esrr@yandex.ru

**RF controlled model based on microcontroller Arduino MEGA 2560\*****A.I. Nezvanov<sup>1</sup>, I.I. Moskvichev<sup>2</sup>, E.D. Zharov<sup>3</sup>**

<sup>1</sup> Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, student of Novosibirsk State Technical University. E-mail: nezvanov@mail.ru

<sup>2</sup> Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, student of Novosibirsk State Technical University. E-mail: ivan-moskvichev96@mail.ru

<sup>3</sup> Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, student of Novosibirsk State Technical University. E-mail: edar.vih.esrr@yandex.ru

For creation of this project we were inspired by last competitions in a robotics in NSTU. Small, but clever robots could walk, fight, go and transport freights on the established routes and obstacles. The main idea was to create the machine robot model operated by the microcontroller programmed by us, because about 90% of the robots provided at competitions were created from the Lego Mindstorms sets. We considered that it will be much more interesting to make the unique car which is completely controlled by our algorithm. At the moment the broadcast and wireless technologies strongly entered our life. Many, despite active progress and a century of big information, have even no idea how these systems and technologies work. One of the purposes in this work is to learn to apply wireless technologies to the solution of applied tasks and to collect model for practical use of the received skills. Management of radio-controlled model of the moving rear-wheel car is realized on the Arduino Mega 2560 microcontroller, control signals tranceiver is realized on the Arduino Uno R3 microcontroller and the joystick which is, in fact, two potentiometers connected in one handle. Power supply of model is accumulator of 12 volts for engines, for low-voltage chains (microcontroller) the semiconductor voltage stabilizer on 5 volts, and for the controller of management the usb-socket from the computer. We sincerely hope that this material will be useful to the beginning students, who wish to study and apply this technology on the radio-controlled devices. In the future it is planned to develop the project and to use this technology in the experimental and scientific purposes. This material is intended only in the fact-finding purposes.

**Keywords:** microcontroller, Arduino Mega 2560, robot technologies, motorShield, servo, transceiver, Serial Parallel Interface, radio control, programming

DOI: 10.17212/2307-6879-2016-2-88-98

**REFERENCES**

1. Banzi M. *Getting started with Arduino*. Sebastopol, CA, Maker Media, 2011. 128 p.
2. Brian W.E. *Arduino programming notebook*. San Francisco, CA, Creative Commons, 2007. 35 p.
3. McRoberts M. *Beginning Arduino*, New York, NY, Apress, 2010. 459 p.

---

\* Received 15 March 2016.

4. Sommer U. *Arduino. Mikrocontroller-programmierung mit Arduino/Free-duino*. Poing, Franzis, 2010. 264 p. (Russ. ed.: Sommer U. *Programmirovaniye mikrokontrollernykh plat Arduino/Freeduino*. St. Petersburg, BHV-Peterburg Publ., 2012. 256 p.).
5. Dnishchenko V.A. *Distsionnoye upravleniye modelyami* [Radio control of models]. St. Petersburg, Nauka i tekhnika Publ., 2007. 456 p.
6. Boxall J. *Arduino workshop*. San Francisco, No starch press, 2013. 394 p.
7. Banzi M. *Getting started with Arduino*. Beijing, O'Reilly, 2008. 118 p. (Russ. ed.: Bantsi M. *Arduino dlya nachinayushchikh volshebnykh*. Moscow, Rid Grupp Publ., 2012. 128 p.).
8. Igoe T. *Making things talk*. Sebastopol, Maker Media, 2011. 470 p. (Russ. ed.: Igo T. *Arduino, datchiki i seti dlya svyazi ustroystv*. St. Petersburg, BHV-Peterburg Publ., 2015. 544 p.).
9. Monk S. *Programming Arduino: getting started with sketches*. 2<sup>nd</sup> ed. New York, McGraw-Hill Education, 2016. 176 p. (Russ. ed.: Monk S. *Programmirovaniye Arduino: osnovy raboty so sketchami*. St. Petersburg, Piter Publ., 2016. 176 p.).
10. Petin V. *Proekty s ispol'zovaniem kontrollera Arduino* [Projects using arduino]. St. Petersburg, BHV-Peterburg Publ., 2014. 400 p.
11. Margolis M. *Arduino cookbook*. Sebastopol, CA, O'Reilly Media, 2012. 724 p.
12. Petin V. *Proekty s ispol'zovaniem kontrollera Arduino*. [Projects using arduino]. 2<sup>nd</sup> ed. St. Petersburg, BHV-Peterburg Publ., 2015. 464 p.
13. Blum J. *Exploring Arduino: tools and techniques for engineering wizardry*. Indianapolis, Wiley and Sons, 2013. 357 p.
14. Boxall J. *Arduino Workshop: a hands-on introduction with 65 projects*. Kindle ed. San Francisco, No Strach Press, 2008. 250 p.
15. Savasgard E. *Arduino: 101 beginners guide: how to get started with your Arduino (tips, tricks, projects and more!)*. Kindle ed. Sebastopol, CA, O'Reilly Media, 2008. 38 p.