

УДК 681.522

МОДЕРНИЗАЦИЯ УСТРОЙСТВА УПРАВЛЕНИЯ МАНИПУЛЯТОРОМ МП-9С*И. Ю. Кузнецов, В. А. Детистов, А. Ю. Зайцев*

Донской государственной технической университет (г. Ростов-на-Дону, Российская Федерация)

Рассмотрено ЭЦПУ-6030 — многофункциональное устройство управления манипуляторами. Оно было разработано и производилось в конце прошлого века. В настоящее время морально устарело. В статье рассмотрены вопросы модернизации ЭЦПУ-6030 с сохранением функциональных возможностей, но уменьшением удельной энергоёмкости и габаритов устройства. Представлена принципиальная электрическая схема модернизированного устройства, центральной частью которой является контроллер семейства AVR — ATmega328. Предложена программа управления контроллером.

Ключевые слова: манипулятор, МП-9С, устройство управления, контроллер AVR, модернизация ЭЦПУ-6030, Arduino NANO.

UPGRADING THE MANIPULATOR CONTROL MP-9S*I. Yu. Kuznetsov, V. A. Detistov, A. Yu. Zaytsev*

Don State Technical University (Rostov-on-Don, Russian Federation)

The paper considers ETSPU-6030, which is a multifunctional manipulator control device. It was developed and produced at the end of the last century. It is currently obsolete. The article deals with the issues of modernization of ETSPU -6030 with the preservation of functional capabilities, but reducing the specific energy intensity and dimensions of the device. The basic electrical diagram of the upgraded device is presented, the central part of which is the controller of the AVR family — ATmega328. A program for controlling the controller is proposed.

Keywords: manipulator, MP-9S, control device, AVR controller, ETSPU-6030 modernization, Arduino NANO.

Введение. Робот МП-9С производства АО «АВТОВАЗ» применяется на сборочных, транспортировочных операциях, а также при обслуживании прессов холодной штамповки [1]. Пневматический робот имеет три степени свободы. Электронное цикловое программное устройство ЭЦПУ-6030 разрабатывалось и применяется для управления МП-9С, однако оно имеет ряд недостатков: морально устарело, сложно в освоении, не позволяет подключиться к ПК.

Цель представленной работы — создание устройства в соответствии с прототипом ЭЦПУ-6030, с меньшими габаритами и удельной энергоёмкостью, а также функциональными возможностями, не уступающими прототипу.

Основная часть

Описание устройства. В манипуляторе звенья двигаются не дискретно относительно сочленения, а достигают крайних положений — и такие случаи отмечаются сигналом от геркона. Пневматические распределители и герконы питаются от постоянного напряжения в 24 вольта. Один распределитель потребляет до одного ампера [2]. Из-за конструктивных особенностей робота число одновременно задействованных распределителей не может превышать четырех. Потребление тока герконом составляет 0,1 А, а количество одновременно задействованных — не более трех. Принимая во внимание описанные выше характеристики электропитания элементов робота, блок питания должен иметь следующие выходные характеристики: 24 вольта постоянного тока; ток от 4,4 А.

Устройство управления выполняет следующие операции: запись цикла работы, работа по циклу, единичные команды, экстренная остановка. Такой функционал при минимальных габаритах может обеспечить контроллер. В качестве контроллера выбран Arduino NANO на ATmega328 семейства AVR. Его основные преимущества: компактность (в сравнении с другими решениями Arduino), достаточность программируемых выводов и простота среды разработки на основе языка С. Питание всех элементов робота осуществлялось с устройства управления (эту функцию возьмет на себя разрабатываемое устройство). Контроллер работает с сигналами, напряжение которых не превышают 5 вольт, а ток на одном выводе — не более 0,4 А, что меньше значений, необходимых для робота. В комплекте должны быть внешний блок для питания устройства управления и робота, а также элементы коммутации.

В качестве элементов коммутации выступают силовые ключи на основе MOSFET транзистора IRF520 [3] со следующими характеристиками:

- управляющее напряжение 3,3–5 В,
- напряжение коммутации нагрузки 0–24 В,
- выходной ток нагрузки до 5 А.

Использование MOSFET-транзистора обусловлено наличием гальванической развязки между затвором и стоком (истоком), что защитит выводы сигнальных линий контроллера от мощных силовых линий. Герконы подключаются к выводам контроллера через резистивный делитель напряжения.

Коммутация с герконовыми датчиками выполнена через делитель напряжения (рис. 1), так как рабочее напряжение датчиков — 24 вольта, а на выходах контроллера напряжение составляет 5 вольт.

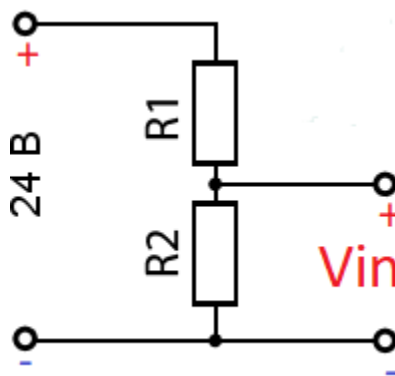


Рис. 1. Условная схема делителя напряжения

Высокий уровень напряжения на выходе контроллера начинается с 4,2 вольт. Номиналы R1 и R2 рассчитаны согласно закону Ома как 420 и 180 ом соответственно.

Необходимое питание контроллеру обеспечивает линейный стабилизатор L7909 [4]. Он понижает напряжение блока питания до уровня, необходимого для нормального функционирования [5] контроллера.

Принимая во внимание все заявленные требования элементов по мощности питания, выбран блок питания LRS-150-24. Он обеспечивает 24 вольта и 6,5 ампер, тогда как устройству управления понадобится до 5 ампер.

Схема устройства изображена на рис. 2.

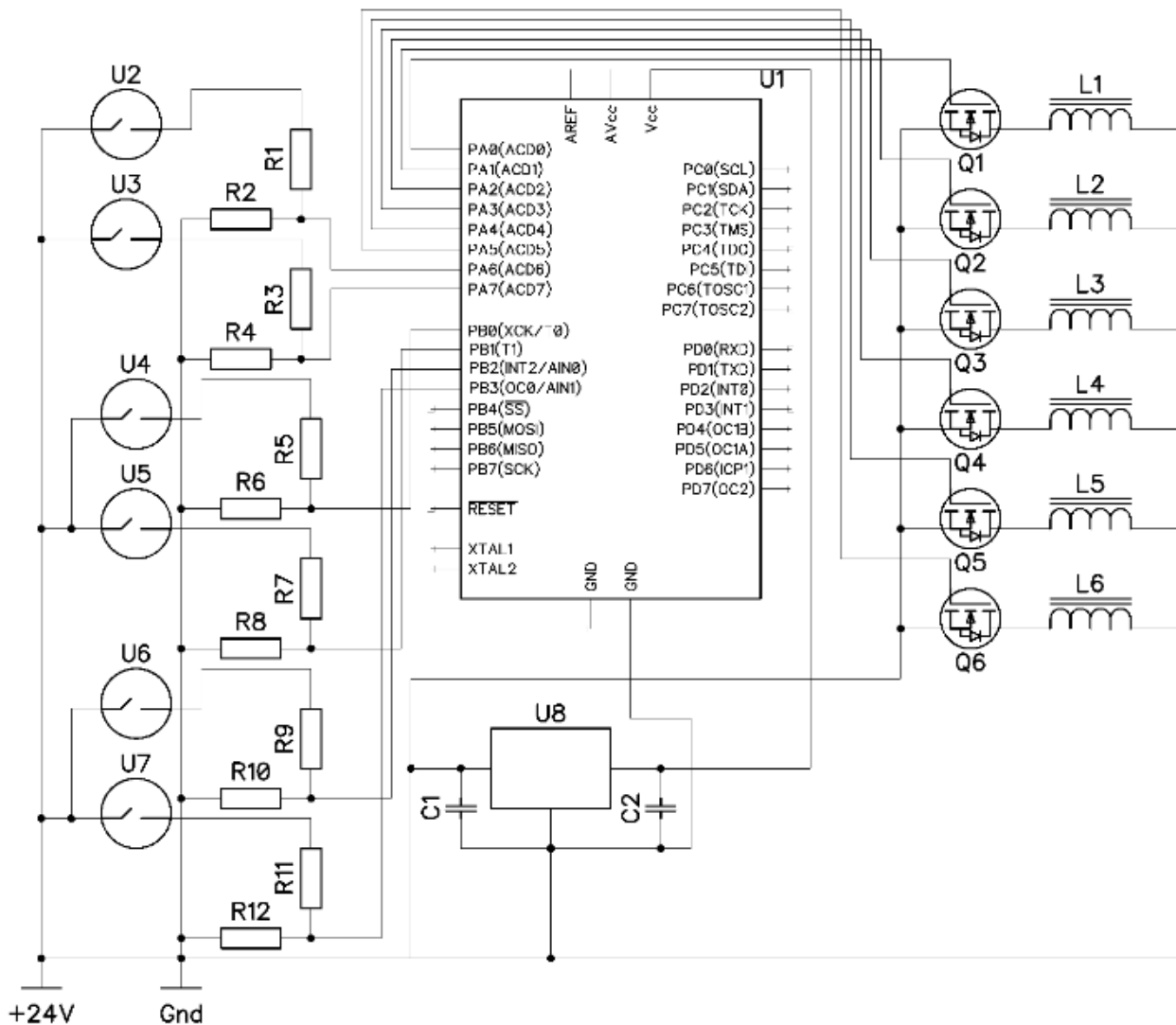


Рис. 2. Принципиальная электрическая схема устройства управления манипулятором МП-9С

L1 — L6 обозначают пневматические распределители. Между транзисторами Q1 — Q6 и распределителями так же, как и между герконами и их резистивными делителями, стоят интерфейсы (разъемы подключения). Интерфейсы манипулятора РТТ2 [6], интерфейсы устройства управления могут быть различны, но при выборе отличного от РТТ2 необходим нестандартный соединительный кабель.

Герконы, детектирующие концевые положения одного штока, подключены общим «+», так как одновременно может быть активен только один из них.

Сглаживающие конденсаторы C1, C2 необходимы для снижения пульсаций на входе контроллера.

Ниже представлена программа, записанная в контроллер через среду Arduino ADE.

Программа 1

```
1 void setup() {
2   Serial.begin(9600); // инициализируем подключение к
                        // ПК со скоростью 9600 бит в
                        // секунду
3   int GerC101 = 2;    //присваиваем имена пинам
4   int GerC102 = 3;    (выводам) к которым подключены
5   int GerC201 = 4;    герконы
6   int GerC202 = 5;
7   int GerC301 = 6;
8   int GerC302 = 7;
9   int RasC101 = 8;
10  int RasC102 = 9;
11  int RasC201 = 10;
12  int RasC202 = 11;
13  int RasC301 = 12;
14  int RasC302 = 13;
15  int VKom;

                        //присваиваем имена выводам к
                        //которым подключены
                        //распределители

                        //создаем переменную в которой
                        //будет храниться выбор команды из
                        //списка меню

16  int Zna;           //значение геркона для выбранного
                        //распределителя

17  Int I;
18  Int m;
19  Int posVKom = [100];
20  pinMode (2; INPUT);
21  pinMode (3; INPUT);
22  pinMode (4; INPUT);
23  pinMode (5; INPUT);
24  pinMode (6; INPUT);
25  pinMode (7; INPUT);
26  pinMode (8; OUTPUT);
27  pinMode (9; OUTPUT);
28  pinMode (10; OUTPUT);
29  pinMode (11; OUTPUT);
30  pinMode (12; OUTPUT);
31  pinMode (13; OUTPUT);
32  }
33 void loop() {
34  Serial.println("Меню управления манипулятором // выводим на экран текст меню
35  'MP-9C'.Выберите номер доступной команды:");
36  Serial.println("1. Опускание вниз");
37  Serial.println("2. Подъем вверх");
38  Serial.println("3. Поворот влево");
39  Serial.println("4. Поворот вправо");
40  Serial.println("5. Выдвижение хвата");
40  Serial.println("6. Прижатие хвата");
```



```
Serial.println("7. Программирование
последовательности цикла");
41 Serial.println("8. Старт цикла");
42 VKom = Serial.available();

43 if (VKom == 1){
44   zna = digitalRead(GerC101);
45   switch (zna) {
46     case 0:
47       digitalWrite(RasC101; HIGH);
48       digitalWrite(RasC102; LOW);
49       break;
50     case 1:
51       Serial.println("Выполнено 'Опускные вниз.'");
52       break;
53   }
54   Serial.println("Выполнено 'Опускные вниз.'");
55 }
56 if (VKom == 2){
57   zna = digitalRead(GerC102);
58   switch (zna) {
59     case 0:
60       digitalWrite(RasC102; HIGH);
61       digitalWrite(RasC101; LOW);
62       break;
63     case 1:
64       Serial.println("Выполнено 'Подъем вверх.'");
65       break;
66   }
67   Serial.println("Выполнено 'Подъем вверх.'");
68 }
69 if (VKom == 3){
70   zna = digitalRead(GerC201);
71   switch (zna) {
72     case 0:
73       digitalWrite(RasC201; HIGH);
74       digitalWrite(RasC202; LOW);
75       break;
76     case 1:
77       Serial.println("Выполнено 'Поворот влево.'");
78       break;
79   }
80   Serial.println("Выполнено 'Поворот влево.'");
81 }
82 if (VKom == 4){
83   zna = digitalRead(GerC202);
84   switch (zna) {
85     case 0:
86       digitalWrite(RasC202; HIGH);
87       digitalWrite(RasC201; LOW);
88       break;
```

```
// Запись номера выбранной
команды
// Поиск необходимой реакции на
полученную от ПК команду
```

```
89     case 1:
90         Serial.println("Выполнено 'Поворот вправо.'");
91         break;
92     }
93     Serial.println("Выполнено 'Поворот вправо.'");
94 }
95 if (VKom == 5){
96     zna = digitalRead(GerC301);
97     switch (zna) {
98     case 0:
99         digitalWrite(RasC301; HIGH);
100        digitalWrite(RasC302; LOW);
101        break;
102    case 1:
103        Serial.println("Выполнено 'Выдвижение
104 захвата.'");
105        break;
106    }
107    Serial.println("Выполнено 'Выдвижение
108 захвата.'");
109 }
110 if (VKom == 6){
111     zna = digitalRead(GerC302);
112     switch (zna) {
113     case 0:
114         digitalWrite(RasC302; HIGH);
115         digitalWrite(RasC301; LOW);
116         break;
117     case 1:
118         Serial.println("Выполнено 'Прижатие захвата.'");
119         break;
120     }
121     Serial.println("Выполнено 'Прижатие захвата.'");
122 }
123 if (VKom == 7){
124     for (I = 0; posVKom[I-1] == !0; I++) {
125         Serial.println("Введите номер команды(не вводите
126 7 и 8).");
127         PosVKom[i] = Serial.available();
128         Serial.println("Ввод принят!");
129     }
130     Serial.println("Ввод последовательности цикла
131 окончен!");
132 }
133 if (VKom == 8){
134     Serial.println("Введите количество повторений
135 записанного цикла.");
136     m = Serial.available();
137     Serial.println("Цикл выполняется!");
138     for (Var n = 0; n == !m ; n++) {
139         for (I = 0; posVKom[I] == !0; I++) {
```

```
136      *
137      }
138      }
139      for (i = 0; i < 100; i++) {
140          PosVKom[I] = 0;
141      }
          Serial.println("Цикл успешно выполнен!
Последовательность цикла удалена");
      }
      else {
          Serial.println("Выберите команду, вписав ее
номер");
      };
142 VKom = 0. //сброс переменной в которой
           //храниться команда из меню
    }
```

Строка 132 со знаком * копирует строки 44–113, за тем исключением, что строки, содержащие «if (VKom == номер команды)», изменятся на «if (PosVKom[I] == номер команды)».

Заключение. Разработанное устройство по функционалу не уступает прототипу и превосходит его по таким характеристикам, как удельное энергопотребление, габариты, ремонтпригодность.

На следующих этапах разработки можно расширить функционал устройства, реализовать удаленное управление и обеспечить анализ его качества. Устройство с незначительными поправками в программе микроконтроллера может быть использовано со всеми манипуляторами, которые управляются ЭЦПУ-6030. Решения, использованные в данной работе, могут быть применены и к другим морально устаревшим устройствам управления манипуляторами.

Библиографический список

1. Злыгостев, А. С. Манипулятор промышленного робота как объект управления / А. С. Злыгостев // Робототехника : [сайт]. — URL: <http://roboticslib.ru/books/item /f00/s00/z0000021 /st022.shtml> (дата обращения: 14.01.2020).
2. Динамические характеристики манипулятора МП-9С // Студенческая библиотека онлайн : [сайт]. — URL: https://studbooks.net/2498974/tovarovedenie/dinamicheskie_harakteristiki_manipulyatora_mp-9s (дата обращения: 14.01.2020).
3. Швед, С. М. Обзор наиболее перспективных изделий силовой электроники / С. М. Швед // Science Time. — 2019. — № 6. — С. 131–134.
4. L7909 линейный стабилизатор напряжения // РадиоЧип : [сайт] / М. М. Орлов. — URL: <http://www.radiochip.ru/cgi-bin/goodra.cgi?act=detail&qui=5&un=154> (дата обращения: 15.01.2020).
5. Сидоренко, Б. Atmel: микроконтроллеры семейств AVR, AVR32 и ARM / Б. Сидоренко // Электроника. Наука. Технология. Бизнес. — 2008. — № 8. — С. 46–53.
6. Электрические соединители, разъемы негерметичные ШП // Техкомплект : [сайт] // ООО «Техкомплект». — URL: <http://www.tehcomplect.spb.ru/connctr6.html> (дата обращения: 15.01.2020).



Об авторах:

Кузнецов Иван Юрьевич, магистрант кафедры «Электротехника и электроника» Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), ReVole1@yandex.ru

Детистов Владимир Анатольевич, доцент кафедры «Электротехника и электроника» Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, доцент, v-detistov@mail.ru

Зайцев Александр Юрьевич, заместитель руководителя ресурсного центра робототехники Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), alexandr22333@mail.ru

Authors:

Kuznetsov, Ivan Yu., master's degree student of the Department of Electrical Engineering and Electronics, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344000, RF), ReVole1@yandex.ru

Detistov, Vladimir A., associate professor of the Department of Electrical Engineering and Electronics, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344000, RF), Cand.Sci., associate professor, v-detistov@mail.ru

Zaytsev, Aleksandr Yu., deputy Head of the Robotics Resource Center, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344000, RF), alexandr22333@mail.ru