

РАЗРАБОТКА ПРОВОДНОГО ПУ ДЛЯ УПРАВЛЕНИЯ МЕХАТРОННОЙ СИСТЕМОЙ

Назаров А.А.

Донской государственной технической университет, г.Ростов-на-Дону

Ключевые слова: робототехника, мехатронный объект, пульт управления, Arduino, электроника.

Аннотация. Цель данной статьи – разработка проводного пульта управления для работы с медицинским комплексом. В процессе работы были затронуты аспекты программирования в среде Arduino IDE, проектирование в программном продукте Proteus для упрощенного представления электрической схемы ПУ. Выполнение данной цели будет необходимо на производстве для устранения проблемы в управлении роботизированными линиями и даже целыми комплексами по производству.

DEVELOPMENT OF WIRED CONTROLLERS FOR CONTROLLING THE MECHATRONIC SYSTEM

Nazarov A.A.

Don state technical university, Rostov-on-don

Keywords: robotics, mechatronic facility, control panel, arduino, electronics.

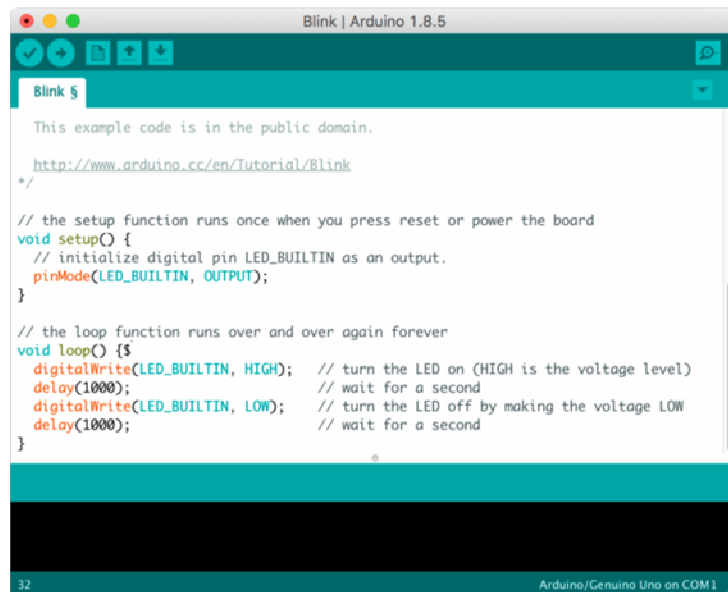
Abstract. The purpose of this article is to develop a wired control panel for working with a medical complex. During the work, aspects of programming in the Arduino IDE environment, designing in the Proteus software product for a simplified representation of the electrical circuit of the control panel were touched. Fulfillment of this goal will be necessary in production to eliminate problems in the management of robotic lines and even entire production complexes.

Была поставлена задача – разработать дистанционный пульт управления мехатронным медицинским реабилитационным комплексом.

Эта задача включала в себя разработку схемы коммутации электронных устройств в среде симуляции, изготовления и программирование прототипа, его настройку. Необходимость разработки пульта управления очень актуальна, поскольку значительно облегчает управление, его можно производить дистанционно.

Для работы был выбран программный продукт Arduino IDE (рис. 1) [1-3], а также микроконтроллер Arduino Nano. Как видно из рисунка выше, Arduino IDE достаточно просто в использовании. Имеется две базовые функции, с помощью которых и будет работать микроконтроллер. Конечно, допускается написание собственных функции, в данном проекте такая возможность необязательна к применению.

Реализация простого пульта управления в программной среде была представлена в двух скетчах. Первый скетч реализован в самой установке, задача которого – это управление редуктором и снятие данных с датчиков Холла и энкодеров (рис. 2).



```

Blink §

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

```

Рис. 1. Arduino IDE



Рис. 2. Внешний вид мехатронного объекта

Второй скетч был реализован непосредственно в пульт управления (листинг программы приведен ниже).

```

Int r,enc_u,skor;
Bool ch1;
#include<LiquidCrystal_I2C.h>
void setup() {
  lcd.LiquidCrystal_I2C(x27);
  pinMode(3,INPUT);//Кнопка смены режима
  pinMode(4,INPUT);//Кнопка 1
  pinMode(5,INPUT);//Кнопка 2
  pinMode(A0,INPUT);//Потенциометр
  Serial.begin(9600); }
void loop() {
  while(Serial.available>0)
  { skor=Serial.read();
  delay(50);
  enc_u=Serial.read();
  lcd.clear();
  lcd.setCursor(0,1);

```

```

lcd.print(«skor»);
lcd.print(skor);
lcd.setCursor(1.1);
lcd.print(«enc_ugl»);
lcd.print(enc_ugl);
delay(50); }
if (digitalRead(3)==1 && !ch1)
{ ch1=true; }
if (digital Read(3)==0 && ch1)
{ ch1=false;
r++;r;
Serial.Write(«r»);
if (r>1) {r=0;} }
if (digitalRead(4)==1)
{ Serial.write(«10»);}
if (digitalRead(5)==1)
{ Serial.write(«-10»);}
delay(50);
Serial,write(analogRead(A0);
lcd.setCursor(1.5);
lcd.print(analogRead(A0)); }

```

Код написан с поправкой на необходимость введение тестового режима, который заключается в медленном перемещении подвижной части установки из одного края в другой.

Далее был задействован программный продукт Proteus (рис. 3), чтобы показать схему подключения к портам ввода и вывода.

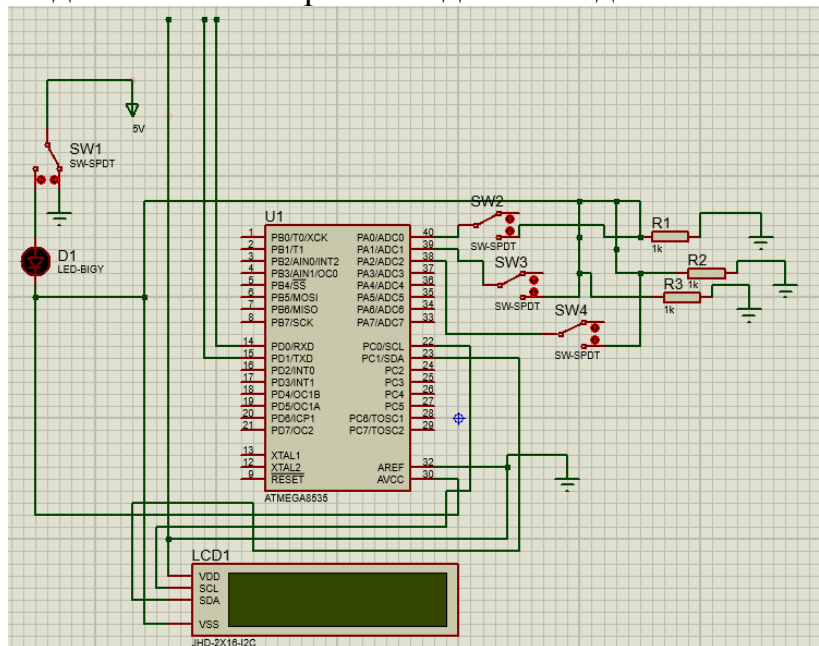


Рис. 3. Схема подключения в программе Proteus

Как видно из рисунка выше, пульт управления представляет собой достаточно простую схему, а также с возможностью мониторинга значения переменных, необходимых в процессе работы.

Далее уже происходит непосредственно сборка по схеме и загрузка написанного кода (рис. 4).



1 - кнопка питания; 2 - LCD-экран; 3 - потенциометр для регулирования скорости движения установки; 4 - кнопка для смены режима; 5,6 – кнопки, отвечающие за положение подвижной части установки; 7 - вывод, идущий непосредственно к установке

Рис. 4. Пульт управления

Таким образом, создание пульта управления облегчило задачу в управлении мехатронным объектом, что несомненно важно в сфере производства, поскольку создание подобных устройств не только экономит средства на рабочей силе, но и позволяет управлять большими объемами работы станков.

Список литературы

1. Монк С. Програмируем Arduino. Профессиональная работа со скетчами. – СПб.: Питер, 2017. – 208 с.
2. Петин В.А., Биняковский А.А. Практическая энциклопедия Arduino – М.: ДМК Пресс, 2017. – 152с.
3. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. – СПб.: БХВ-Петербург, 2016. – 320 с.

Сведения об авторе:

Назаров Александр Александрович – студент, ДГТУ, г.Ростов-на-Дону.