

УДК 629.045

***СИСТЕМА МОНИТОРИНГА ЗА УДАРНЫМИ И ПРОЧИМИ
ВОЗДЕЙСТВИЯМИ НА ОБЪЕКТЫ ТРАНСПОРТИРОВАНИЯ:
АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ***

Брылкин А.А.

магистр 2 курса,

Орловский государственный университет имени И.С. Тургенева,

г. Орел, Россия

Аннотация

В статье описана разработка системы мониторинга за ударными и прочими воздействиями на объекты транспортирования: программное обеспечение системы. Приведен анализ систем мониторинга, а также приведено описание структурной системы мониторинга за ударными и прочими воздействиями.

Была разработана программа управления системой, позволяющая производить сбор данных.

Ключевые слова: система мониторинга, программа управления.

***MONITORING SYSTEM FOR SHOCK AND OTHER IMPACTS ON
THE TRANSPORTATION OBJECTS: SENSOR BLOCK***

Brylkin A. A.

master 2nd year,

Orel state University named after I. S. Turgenev,

Orel, Russia

Annotation

The article describes the development of a system for monitoring impact and other impacts on transportation facilities: system software. The analysis of monitoring systems, as well as a description of the structural system of monitoring of impact and other impacts.

Was developed the system management program that allows to collect data.

Key words: monitoring system, control program.

Система мониторинга за разными воздействиями на объект в настоящее время используют часто и в разных сферах деятельности.

Мониторинг - система наблюдений за процессом (явлением, фактором) природного или техногенного происхождения, состоянием окружающей среды, объекта, а также оценка и прогноз их изменений и развития.

В данной статье разрабатывается программное обеспечение для системы мониторинга за ударными и прочими воздействиями на объекты транспортировки, на базе платформы для разработки с микроконтроллером ATmega32U4 - Iskra Neo.

Объектом исследования является блок обработки измерительной информации. Блок обработки измерительной информации нужен для того, чтобы отслеживать ударные и прочие воздействия на объект. После транспортировки блок обработки измерительной информации должен нести в себе информацию о воздействии удара, вибрации, температуры и влаги, которые произошли в процессе транспортирования.

В качестве предмета исследования рассматриваются различные виды воздействий на объекты транспортирования, которые потенциально снижают надежность изделий. Исходя из классификации приборов по условиям их применения, основными влияющими на объект факторами при транспортировки являются: вибрация, удар, температура и влажность.

Целью исследования является создание системы мониторинга за ударными и прочими воздействиями на объекты транспортирования, которая позволит выявить нарушения в перевозке объекта.

Для достижения данной цели были поставлены и решены следующие задачи:

1. Разработка структурной схемы проектируемого устройства.
2. Разработка схемы подключения.
3. Разработка программного кода для системы управления в среде Arduino IDE.
4. Разработка корпуса для блока обработки измерительной информации.

Ниже приведена структурно-функциональной схемы системы мониторинга (рисунок 1).

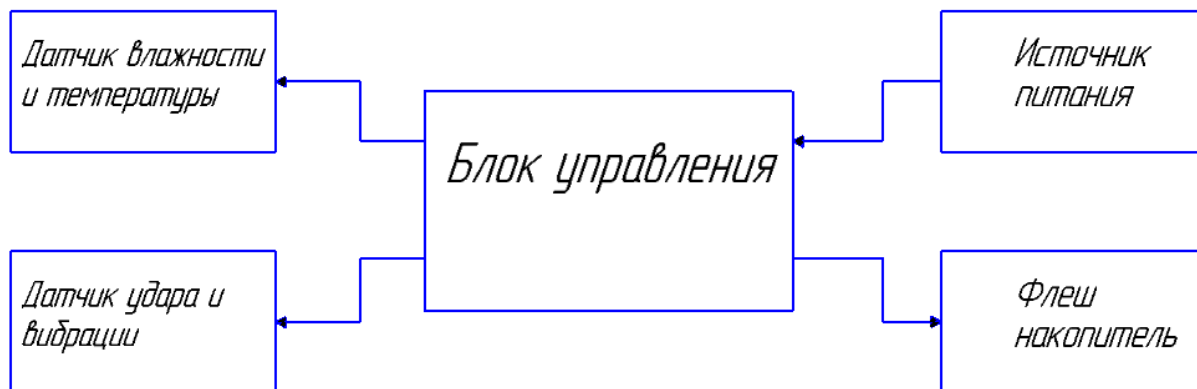


Рисунок 1 – Структурная схема системы

Блок управления представляет собой ISKRANEO -платформу для разработки с микроконтроллером ATmega32U4.

Блок управления является основным звеном структурной схемы. В него поступают сигналы со всех датчиков оборудования (датчика влажности и температуры - DHT22, датчика удара и вибрации - акселерометр 832M1), которые обрабатываются и сравниваются с заданными значениями.

Сигналы с блока управления поступают на флеш накопитель, который представляет собой модуль SD карты с флеш – картой.

Питание блока управления осуществляется батареей 9 В.

При выборе системы мониторинга учитывались параметры, соответствующие следующим критериям:

1. Система мониторинга за ударными и прочими воздействиями на объекты транспортировки должна соответствовать условиям в которых производится перевозка объекта.
2. Дешевизна. Система мониторинга не должна быть дорогой.
3. Система мониторинга за ударными и прочими воздействиями на объекты транспортировки должна обеспечивать запись любых изменений измеряемых величин в течении указанного срока транспортировки.
4. Должна быть обеспечена высокая надежность и безотказность системы, так как в условиях транспортировки невозможно устранить неисправность в случае ее возникновения.
5. Высокая энергоэффективность системы мониторинга, так как установлена автономная система электрообеспечения, ресурсы которой ограничены.
6. Информация об окружающей среде, температуру и влажность.
7. Информацию о воздействиях на объект вибраций и удара.

Учитывая данные критерии, были выбраны следующие комплектующие:

1. Iskra Neo — платформа для разработки с микроконтроллером ATmega32U4.
2. MMA7361, аналоговый трехосный акселерометр.
3. SD card модуль.
4. Датчики температуры и влажности DHT22.
5. Батарея на 9В.

Iskra Neo отвечает за подачу управляющих сигналов на исполнительные механизмы, снятие показаний с датчиков и последующую запись на флеш-накопитель. Микроконтроллер ATmega32U4, на котором базируется контроллер, поставляется с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов по USB интерфейсу. Связь осуществляется оригинальным протоколом STK500.

На рисунке 2 представлена наглядная схема подключения всех элементов системы управления и исполнительных механизмов, реализованная в среде Autodesk Circuits.

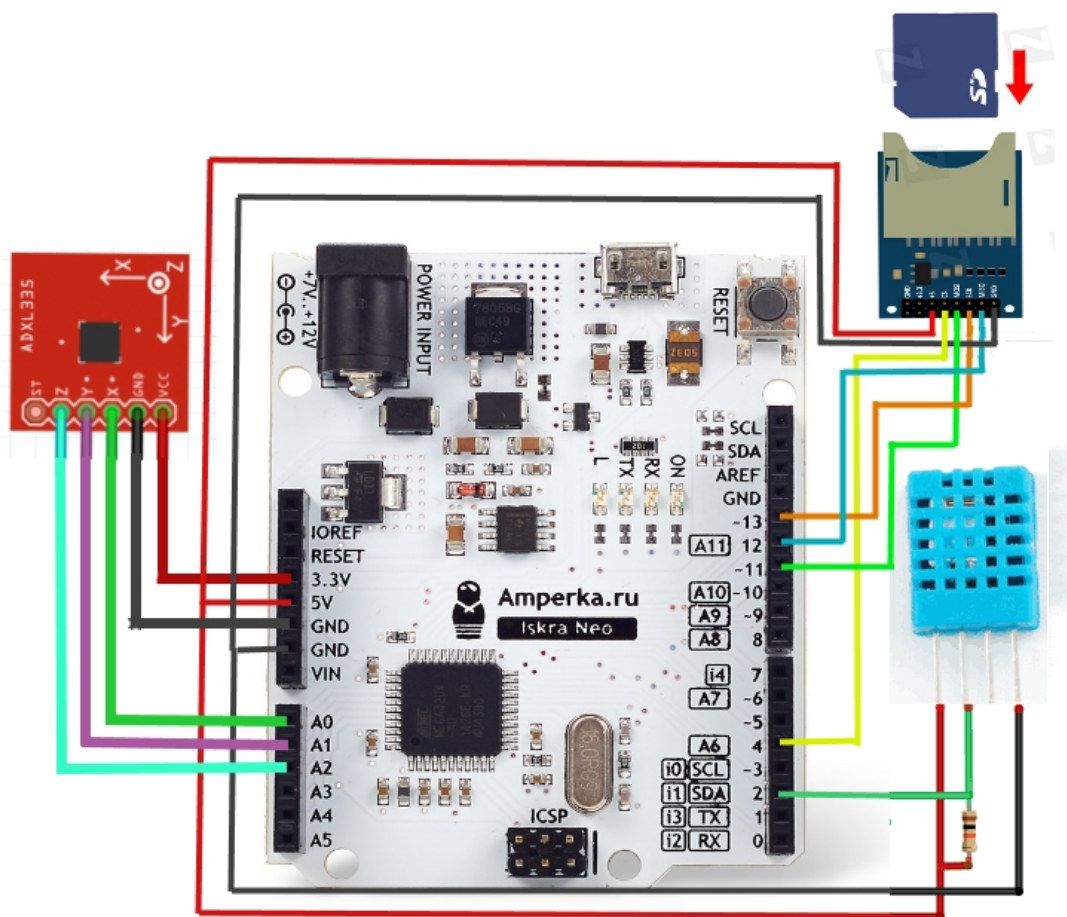


Рисунок 2 – Схема подключения.

Питание всех датчиков осуществляется за счет Iskra Neo напряжением 5 В и 3.3В. Питание Iskra Neo осуществляется от батареи напряжением 9 В. Датчик температуры и влажности DHT22 формирует цифровой сигнал и подает его на цифровой вход вход 2 микроконтроллера. Датчик вибрации MMA7361 формирует аналоговый сигнал и подает его на аналоговые входы А0-А2 микроконтроллера. Модуль SD-карты подключается к цифровым входам 11-13 микроконтроллера.

Так как данная среда предоставляет ограниченный комплект комплектующих, то некоторые элементы могут отличаться от тех, которые установлены на реальном устройстве.

Далее разрабатывается программное обеспечение для системы мониторинга за ударными и прочими воздействиями. Подробно разбирается процессы работы с датчиками. Производится тестирование работы датчиков.

Работа с датчиком DHT 22

Загружается тестовый скетч, который будет отображать влажность и температуру в окне Монитора порта.

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
```

```
    if (isnan(t) || isnan(h)) {  
      Serial.println("Failed to read from DHT");  
    }  
    else {  
      Serial.print("Humidity: ");  
      Serial.print(h);  
      Serial.print(" %\t");  
      Serial.print("Temperature: ");  
      Serial.print(t);  
      Serial.println(" *C");  
    }  
  }  
}
```

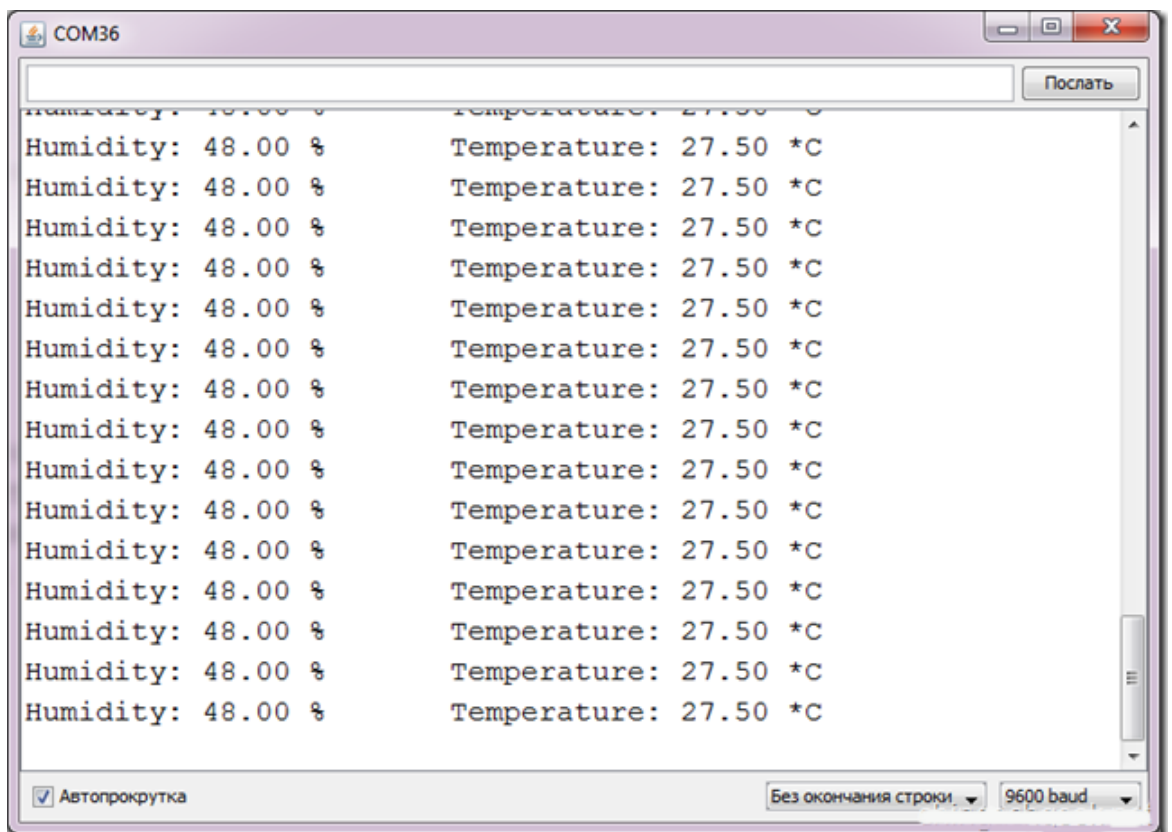


Рисунок 3 – Окно монитора порта.

Производится настройка и калибровка акселерометра.

ISKRANEO имеет 10-разрядный АЦП, а максимальное допустимое напряжение на выводе – 5 вольт. Измеренные напряжения кодируются битами, которые могут принимать только 2 значения – 0 или 1. Это означает, что весь диапазон измерений будет поделён на $(1+1)$ в 10-ой степени, т.е. на 1024 равных отрезка.

Для того чтобы перевести снимаемые показания в вольты, нужно каждое измеренное на аналоговом входе значение поделить на 1024 (отрезка), а затем умножить на 5 (вольт).

Загружаем скетч в память ISKRANEO. Считываем с аналоговых входов показания по трём каналам, преобразовывать их в напряжение и выводить их в последовательный порт.

```
//определяем аналоговые пины:
const int xPin = A0;
const int yPin = A1;
const int zPin = A2;
void setup() {
  //инициализируем послед. порт:
  Serial.begin(9600);
}
void loop() {
  // считываем показания:
  int xRead = analogRead(xPin);
  int yRead = analogRead(yPin);
  int zRead = analogRead(zPin);
  //Выводим показания в порт в Вольтах:
  Serial.print("x: ");
  Serial.print(xRead * 5 / 1024.0);
```


	A	B	C	D	E
	Угол, градусы	Напряжение на выходе Zout, В	Ускорение, g		
1					
2	0	2,03	1,00		
3	90	1,68	0,00		
4	180	1,34	-0,97		
5	270	1,68	0,00		
6	360	2,03	1,00		
7					
8					
9					
10					

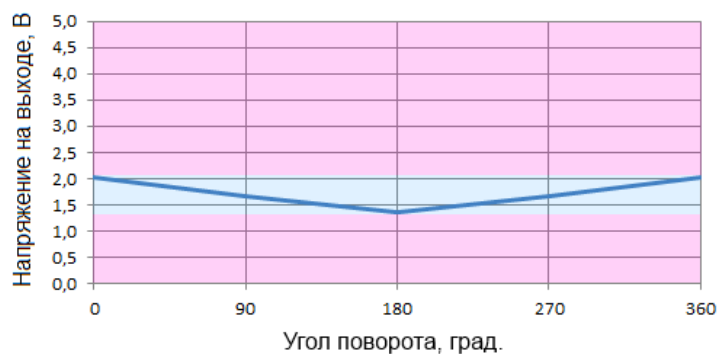


Рисунок 5 – Определение ускорений по трём осям акселерометра.

Для наглядности приведён график напряжений на выходе Z_{out} в зависимости от угла поворота. Голубое поле – это область значений в спокойном состоянии (при ускорении 1g).

При угле поворота 90° на ось Z приходится нулевое ускорение. Т.е. значение 1,67 вольт – это условный ноль Z_0 .

Тогда определим ускорение так: $g = (Z_{out} - Z_0) / S_z$, здесь Z_{out} – измеренное значение в милливольтгах, Z_0 – значение при нулевом ускорении в милливольтгах, S_z – чувствительность датчика по оси Z, измеренная в мВ/g.

Чувствительность акселерометра приведена в паспорте, но лучше провести калибровку акселерометра и вычислить значение чувствительности по формуле: $S_z = Z(0^\circ) - Z(90^\circ)$ В данном случае чувствительность акселерометра по оси Z = $2,03 - 1,68 = 0,35$ В/g.

Аналогично чувствительность считаем для осей X и Y.

В столбце "C" таблицы приводится расчётное ускорение при чувствительности, равной 350 мВ/g. Как видно, расчёты практически совпадают с номинальными величинами, которые даются на первом рисунке из паспорта на датчик, т.е. наш датчик довольно точно показывает свою ориентацию в пространстве.

Ниже приведена программа для системы мониторинга за ударными и прочими воздействиями объекты транспортирования.

```

#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);
#include <SD.h>
const int xPin = A0; //определяем аналоговые пины,
const int yPin = A1; //к которым подключим
const int zPin = A2; //три канала акселерометра
const float Vmax = 5.0; //макс. допустимое напряжение на аналоговом
входе
const float x0 = 1.71; //значения по осям при нулевых "g";
const float y0 = 1.69;
const float z0 = 1.68;
const float sens_x = 0.35; //чувствительность по осям в В/g;
const float sens_y = 0.35;
const float sens_z = 0.35;
File myFile;
void setup() {
Serial.begin(9600);
Serial.println("DHTxx test!");
dht.begin();
Serial.print("Initializing SD card...");
pinMode(10, OUTPUT);
if (!SD.begin(10)) {
Serial.println("initialization failed!");
return;
}
Serial.println("initialization done.");
myFile = SD.open("test.txt", FILE_WRITE);

```

```

if (myFile) {
  Serial.print("Writing to test.txt...");
  myFile.println("zapis temperature i vlagnosti");
  // закрываем файл:
  myFile.close();
  Serial.println("done.");
} else {
  // а если он не открылся, то выводим сообщение об ошибке:
  Serial.println("error opening test.txt");
}
// откроем файл снова, для чтения:
myFile = SD.open("test.txt");
if (myFile) {
  Serial.println("test.txt:");
  // читаем из файла, пока не достигнем конца файла:
  while (myFile.available()) {
    Serial.write(myFile.read());
  }
  // закрываем файл:
  myFile.close();
} else {
  // а если не открылся, то пишем об ошибке:
  Serial.println("error opening test.txt");
}
}

void loop() {
  unsigned int value_x = analogRead(xPin); // считываем значения с
акселерометра
  unsigned int value_y = analogRead(yPin);
  unsigned int value_z = analogRead(zPin);

```

```

float Gx=(value_x*Vmax/1024.0 - x0)/sens_x; //определяем ускорения по
осям
float Gy=(value_y*Vmax/1024.0 - y0)/sens_y;
float Gz=(value_z*Vmax/1024.0 - z0)/sens_z;
delay(500);
float h = dht.readHumidity();
float t = dht.readTemperature();
if (isnan(t) || isnan(h)) {
Serial.println("Failed to read from DHT");
}
myFile = SD.open("test.txt", FILE_WRITE);
// если файл нормально открылся, запишем в него:
if (myFile) {
Serial.print("Writing to test.txt...");
myFile.println("Humidity: ");
myFile.println(h);
myFile.println(" %\t");
myFile.println("Temperature: ");
myFile.println(" *C");
myFile.println("Gx:\t" + String(Gx)); //выводим ускорения
myFile.println("\t| Gy:\t" + String(Gy));
myFile.println("\t| Gz:\t" + String(Gz));
myFile.println("x:\t" + String(round(angle_x))); //выводим углы поворота
акселерометра
myFile.println("o\t| y:\t" + String(round(angle_y)));
myFile.println("o\t| z:\t" + String(round(angle_z)) + "o");
myFile.println();
// закрываем файл:
myFile.close();
Serial.println("done.");

```

```

} else {
// а если он не открылся, то выводим сообщение об ошибке:
Serial.println("error opening test.txt");
}
}

```

В заключении данной статьи разрабатывается корпус блока обработки измерительной информации (рисунок 6).

Чертеж делается для оргстекла 4 мм под лазерную резку. Количество пазов выбиралось из расчета, что бы длины паза была в пределах 20 - 40 мм. Были разработаны два отверстия для крепления полукорпуса с датчиками.

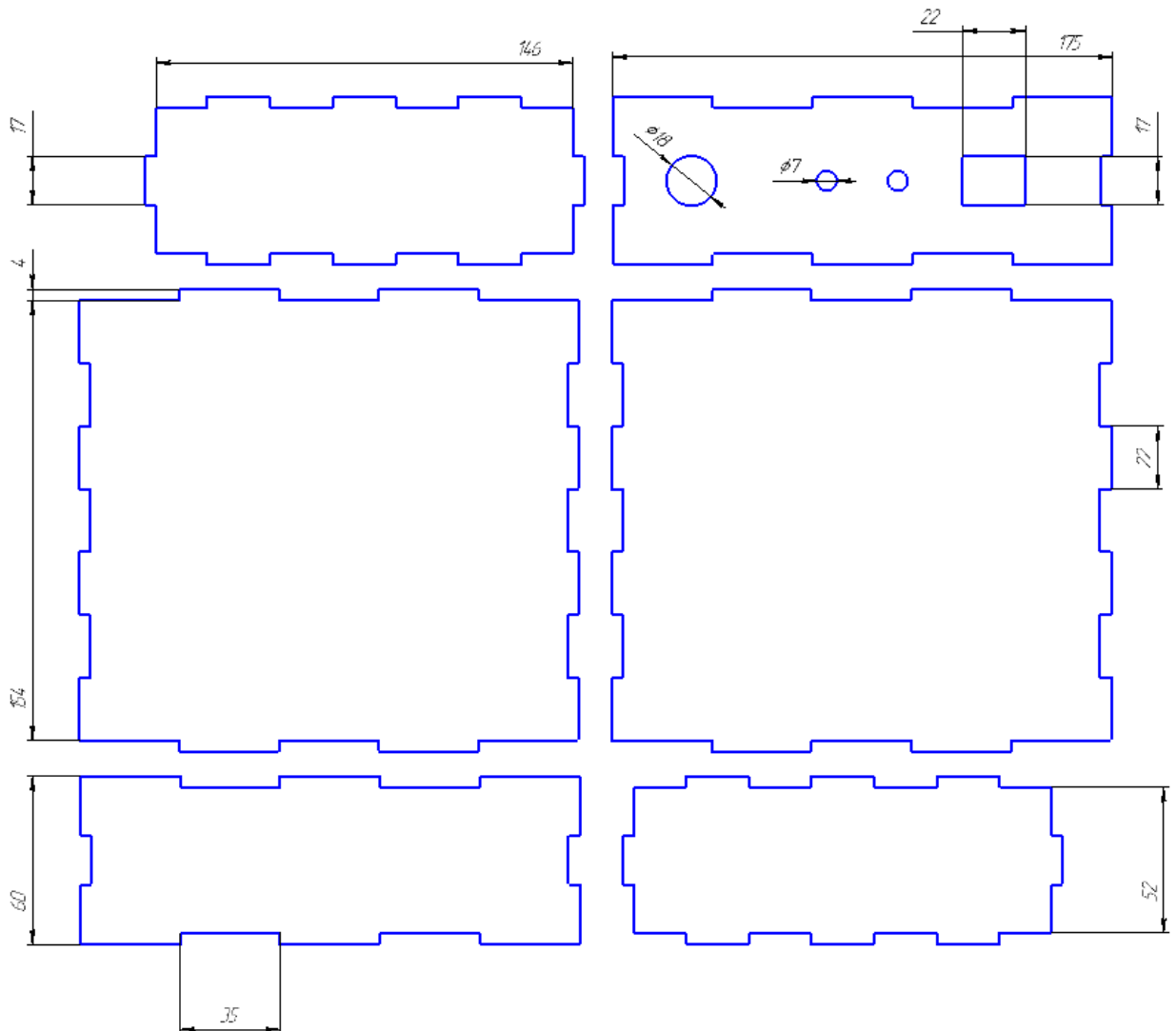


Рисунок 6 – Чертеж корпуса.

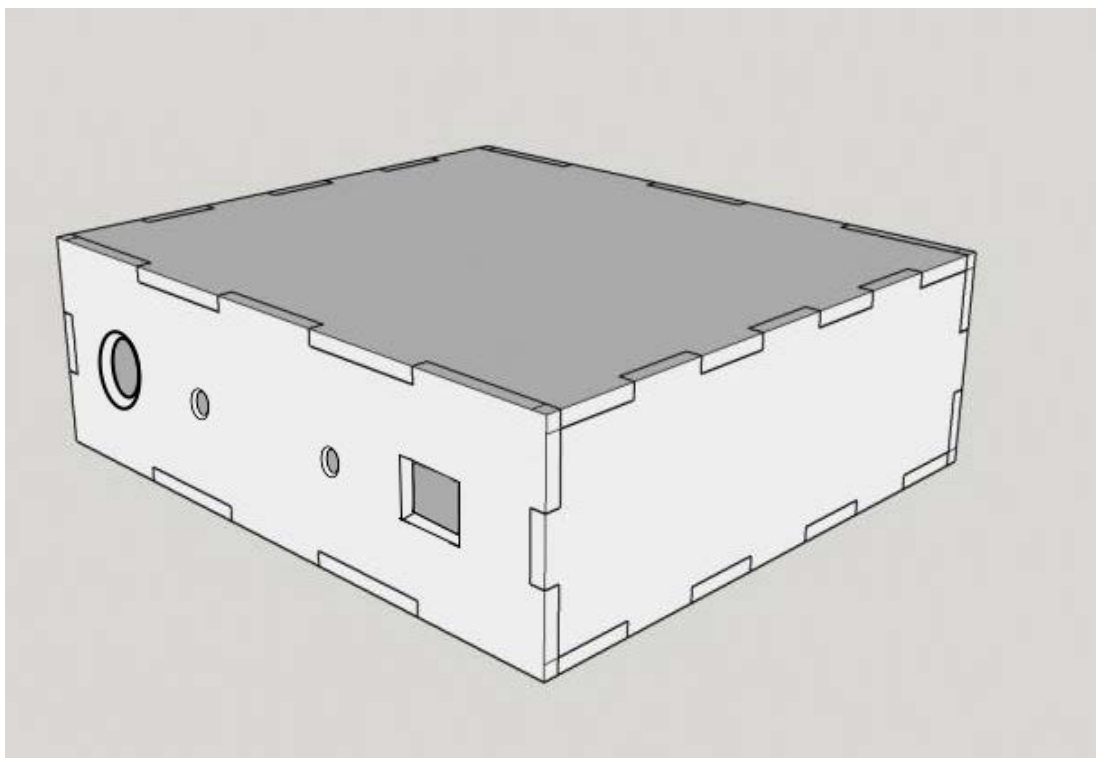


Рисунок 7 – 3D модель корпуса.

Библиографический список

1. Ю. В. Ревич Азбука электроники. Изучаем Arduino - М.: АСТ, 2017.- 226 с.
2. Саймон Монк Програмируем с Arduino. Профессиональная работа со скетчами. М.: Питер, 2017 – 272 с.
3. Торо Карвинен, Киммо Карвинен, Вилле Валтокари Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi.- М.: Вильямс, 2015.- 256 с.