

Решение практики заключительного этапа Аэрокосмос
Продуктовый сектор
Московской предпрофессиональной олимпиады

«Разработка малого космического аппарата и наземной станции»

Этап №1. Прием и расшифровка сообщений

Пункт 1.1 Подключение радиомодуля

Участник должен продемонстрировать собранные устройства, представляющие собой платы Arduino Uno, к которому верно подключены радиомодули nrf24l01+pa+lna. Пример верного подключения показан на рисунке 1.

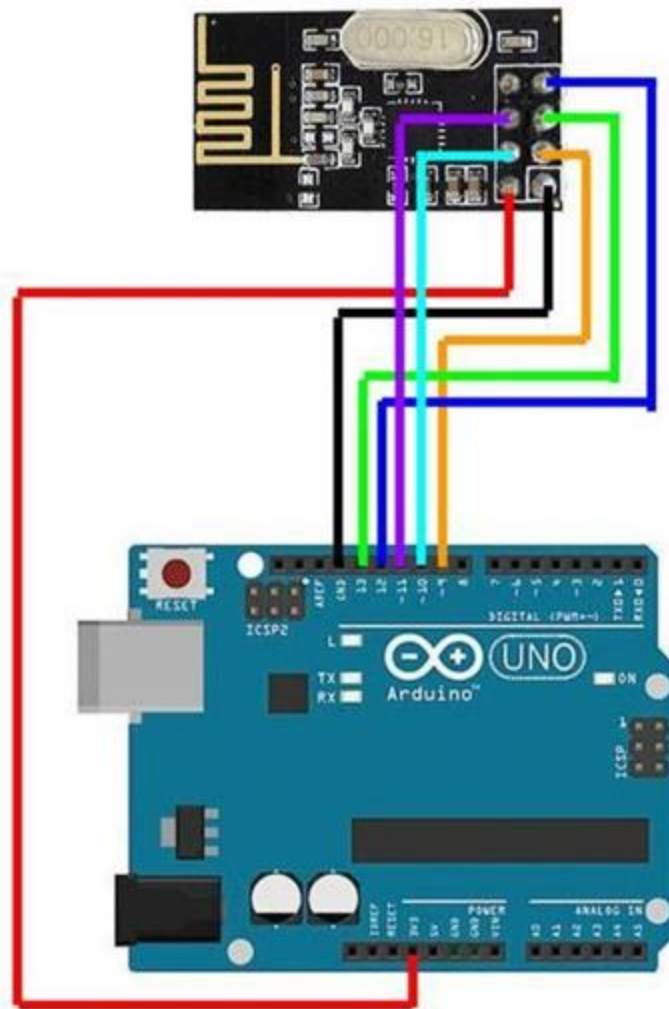


Рис. 1. Корректное подключение модуля NRF24L01 к Arduino Uno

Важно! Провода CE и CSE (на рисунке бирюзовый и оранжевый, второй ряд снизу) радиомодуля могут быть подключены к любому свободному цифровому выходу Arduino (на рисунке верхний ряд выходов на Arduino) кроме 0 и 1 (цифровые выходы с 2 по 10), такое подключение тоже верно. Также черный провод может быть подключен к двум другим выходам на Arduino с маркировкой GND.

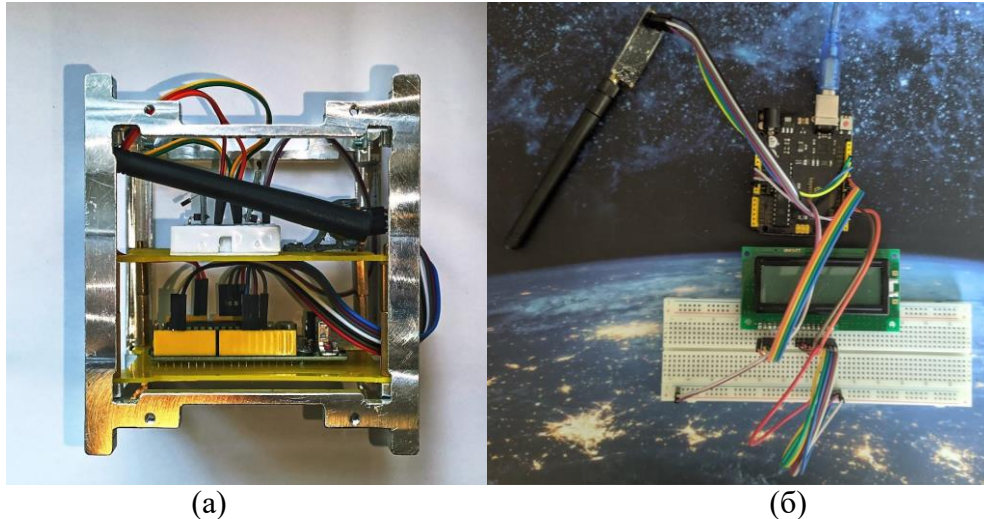


Рис. 2. Идеальный образец, (а) - прототип МКА, (б) - прототип наземной станции

На рис. 2 (а), (б) показаны примеры собранных прототипов МКА (а) и наземной станции (б).

Пункт 1.2 Код приема, дешифровки и вывода сообщений

Правильным является присутствие в коде **прототипа МКА (1.2a-1.2e)** и коде **прототипа наземной станции (1.2f)** фрагментов, выделенных в примере красным. Их отсутствие или орфографические ошибки в них считаются **ошибкой**. Также в коде участника присутствуют фрагменты, выделенные зелёным. Их отсутствие или орфографические ошибки в них считаются **недочетом**.

Пункт 1.2a. Функция приема сообщения с помощью радиомодуля

Участник демонстрирует программный код, который должен выглядеть следующим образом:

Объявление переменных и объектов

Пример 1

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

RF24 radio(2, 3);
const byte receivingAddress[6] = "00000";
const byte forwardingAddress[6] = "00001";

void setup() {
  radio.begin();
  radio.openReadingPipe(1, receivingAddress);
  radio.openWritingPipe(forwardingAddress);
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(0x58);
  radio.startListening();
}
```

Код функции

Пример 1

```
char text[32] = "";  
if (radio.available()) {  
    radio.read(&text, sizeof(text));  
}
```

Пример 2

```
char text[32] = "";  
if (radio.available()) {  
    radio.read(&text, 32);  
}
```

Пункт 1.2b. Функция вывода сообщения в закодированном виде

Участником разрабатывается функция, которая выводит полученные сообщения в закодированном виде в последовательный порт. Участник демонстрирует программный код, который должен выглядеть следующим образом:

Пример 1.

```
void setup(){  
    Serial.begin(9600);  
}  
void printMessage(){  
    Serial.print(text);  
}
```

Пункт 1.2c. Функция декодирования сообщения

Участником разрабатывается функция, которая производит декодирование полученного сообщения. Участник демонстрирует программный код, который должен выглядеть следующим образом:

Пример 1.

```
String decryptAffine(const char* c){  
    String out = "";  
    int a = 3;  
    int b = 7;  
    int m = 26;  
    while(*c){  
        if(*c >= 'A' && *c <= 'Z'){  
            int x = *c - 'A' - b;  
            while(x % a) x += m;  
            out += char(x / a + 'A');  
        }  
        else if(*c >= 'a' && *c <= 'z'){  
            int x = *c - 'a' - b;  
            while(x % a) x += m;  
            out += char(x / a + 'a');  
        }  
        else out += *c;  
        c++;  
    }  
    return out;  
}
```

Пункт 1.2d. Функция вывода декодированного сообщения

Участником разрабатывается функция, которая выводит полученные декодированные сообщения в последовательный порт. Участник демонстрирует программный код, который должен выглядеть следующим образом:

Пример 1.

```
void setup(){  
  Serial.begin(9600);  
}  
void decoder(){  
  Serial.print(message);  
}
```

Пункт 1.2e. Функция передачи сообщения на наземную станцию по радиосвязи

Участником разрабатывается функция, позволяющая передавать сообщение на прототип наземной станции посредством радиосвязи. Участник демонстрирует программный код, который должен выглядеть следующим образом:

Пример 1.

```
radio.stopListening();  
radio.write(&message, sizeof(message));  
radio.startListening();
```

Пример 2.

```
radio.stopListening();  
radio.write(&message, 32);  
radio.startListening();
```

Пункт 1.2f. Реализация вывода сообщения на дисплей на наземной станции

В данном пункте оценивается реализация вывода сообщения на дисплей на прототипе наземной станции. Программный код наземной станции должен содержать следующие строки для приема сообщения по радиосвязи и вывода сообщения на дисплей:

Пример 1.

```
if (radio.available()) {  
    char text[32] = "";  
    radio.read(&text, sizeof(text));  
    lcd.print();  
    lcd.setCursor(0, 1);  
    lcd.print(text);  
}
```

Пример 2.

```
if (radio.available()) {  
    char text[32] = "";  
    radio.read(&text, 32);  
    lcd.print();  
    lcd.setCursor(0, 1);  
    lcd.print(text);  
}
```

Пункт 1.3 Качество декодирования

В данном пункте оценивается верность исполнения программного кода. Проверяющий просит участника включить или перезапустить плату нажатием на кнопку reset и продемонстрировать выводимые кодом сообщения в последовательный порт (окно Arduino IDE на экране конструктора). В последовательный порт должны выводиться следующие сообщения:

Примеры вывода сообщения (нерасшифрованные и расшифрованные)	
[F] ZAF [V] HJU [N] NR Jtn 4 [V] MCF QJP [N] KC WJZ	[I] GPI [W] ASN [C] CM Sec 4 [W] THI DSU [C] BH FSG

[N] QOW QNH
[N] NX
[N] NOH

[C] DLF DCA
[C] CO
[C] CLA

Выделенные красным строки символов - зашифрованные методом Аффинного шифрования сообщения

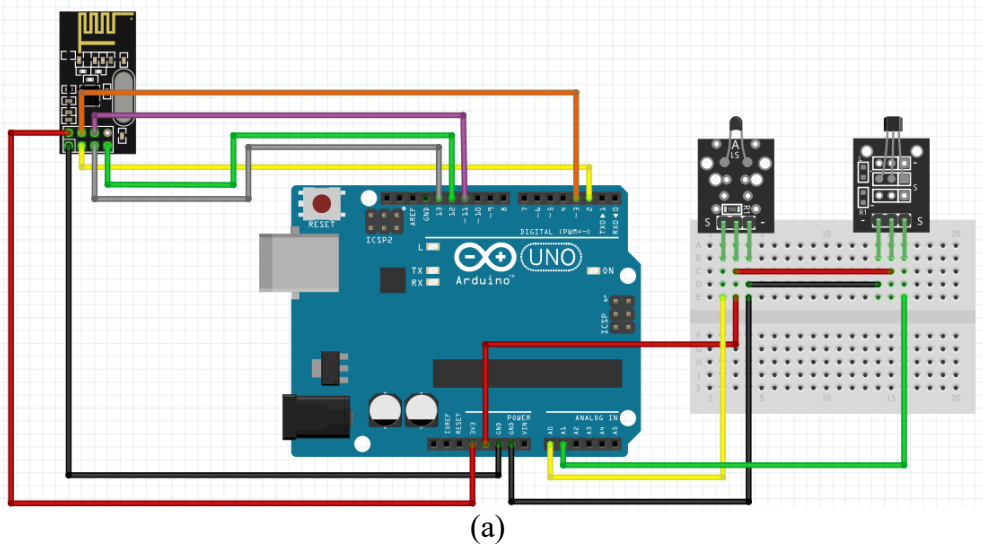
Выделенные зелёным строки символов - дешифрованные сообщения.

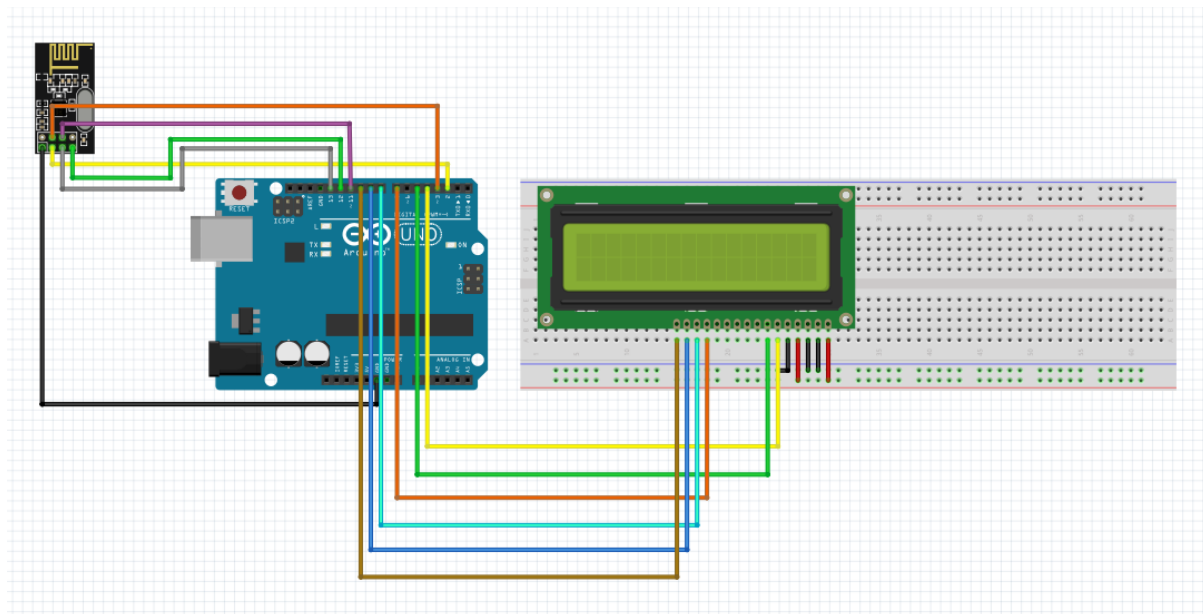
Так как сообщения передаются непрерывно, то порядок сообщений у участников может быть другим.

Этап №2. Сбор и отправка телеметрии

Пункт 2.1 Правильное подключение датчиков к МКА и дисплея к наземной станции

Участник должен продемонстрировать собранное устройство, представляющее собой прототип МКА (плата Arduino Uno, датчик холо KY-035, датчик температуры KY-013, радиомодуль NRF24L01) и прототип наземной станции (плата Arduino Uno, радиомодуль NRF24L01, ЖК-дисплей MT-16S2H). Пример верного подключения модулей и датчиков показан на рис. 3.





(б)

Рис. 3. Пример подключения, (а) - схема прототипа МКА, (б) - схема прототипа наземной станции

Датчики магнитного поля (Холла) и температуры имеют 3 ножки: ножки “+” или Vcc(в зависимости от маркировки) соединяется с красным проводом на схеме, а “-” или GND с черным, ножки S или A0 (на схеме подключены оранжевыми проводами) могут быть подключены к любым свободным аналоговым выходам платы Arduino Uno по выбору участника.

Пункт 2.2

В данном пункте правильным является присутствие в коде **прототипа МКА (2.2а, 2.2б)** и **прототипа наземной станции (2.2в)** фрагментов, выделенных в примере красным. Их отсутствие или орфографические ошибки в них считаются **ошибкой**. Кроме этого, оценивается присутствие в коде участника фрагментов, выделенных зелёным. Их отсутствие или орфографические ошибки в них считаются **недочетом**.

Пункт 2.2а. Функция снятия и вычисления показаний датчиков на МКА

Участником разрабатывается функция для прототипа МКА, которая снимает показания с датчиков, записывает их в заранее созданные переменные. Функция для датчика температуры должна переводить получаемое значение с АЦП в градусы цельсия. Участник демонстрирует программный код, который должен выглядеть следующим образом:

```
#define KELVIN_ZERO 273.15
#define T0 float(25 + 273.15)
#define R0 (float)10000
#define B (float)3950

void setup() {
    inductance0 = analogRead(PIN_MAGNET);
}

float getTemperature() {
    float value = analogRead(PIN_TEMP);
    float R = R0 / (1023 / (float)analogRead(0) - 1);
    float temp = 1 / (1 / T0 + (1 / B) * log(R / R0));
```

```
temp -= KELVIN_ZERO;  
return temp;  
}
```

```
int getMagnet() {  
int inductance = analogRead(PIN_MAGNET) - inductance0;  
return inductance;  
}
```

Пункт 2.2б. Функция передачи показаний датчиков на наземную станцию по радиосвязи

Участник демонстрирует программный код прототипа МКА, который должен выглядеть следующим образом:

Пример 1.

```
radio.stopListening();  
radio.write(&sensor_data, sizeof(sensor_data));  
radio.startListening();
```

Пример 2.

```
radio.stopListening();  
radio.write(&sensor_data, 32);  
radio.startListening();
```

Пункт 2.2в. Функция вывода показаний датчиков на LCD дисплей на наземной станции

Участник демонстрирует программный код наземной станции, который должен выглядеть следующим образом

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(PIN_RS, PIN_EN, PIN_DB4, PIN_DB5, PIN_DB6, PIN_DB7);
```

```
void printSensorData() {  
lcd.begin(16, 2);  
lcd.setCursor(0, 1);  
lcd.print("t=");  
lcd.print(temp);  
lcd.print(" m=");  
lcd.print(inductance);  
}
```

Пункт 2.3 Работоспособность прототипов МКА и наземной станции

В данном пункте оценивается верность исполнения программного кода. Проверяющий просит участника включить или перезапустить плату нажатием на кнопку reset и продемонстрировать работу устройства.

Прототип наземной станции должен выводить показания всех датчиков (температура и магнитная индукция). Актуальная информация с датчика должна выводиться на дисплей не реже чем раз в 5 секунд.

Этап №3. 3D-моделирование корпуса МКА формата Cubesat 1U

Участник должен продемонстрировать трехмерную модель рамы, а также трехмерные модели модулей и датчиков, разработанные в соответствии с чертежами.

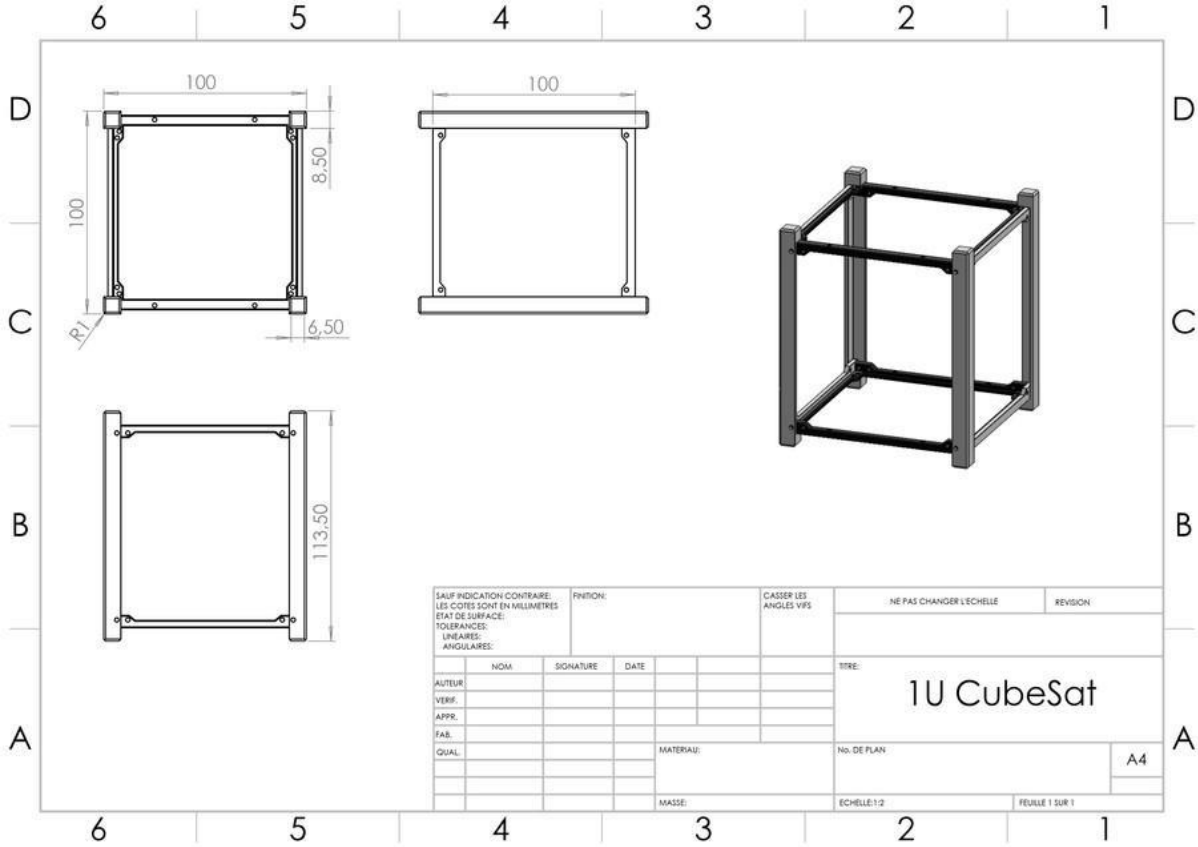


Рис. 4. Чертеж рамы, предоставленный участникам

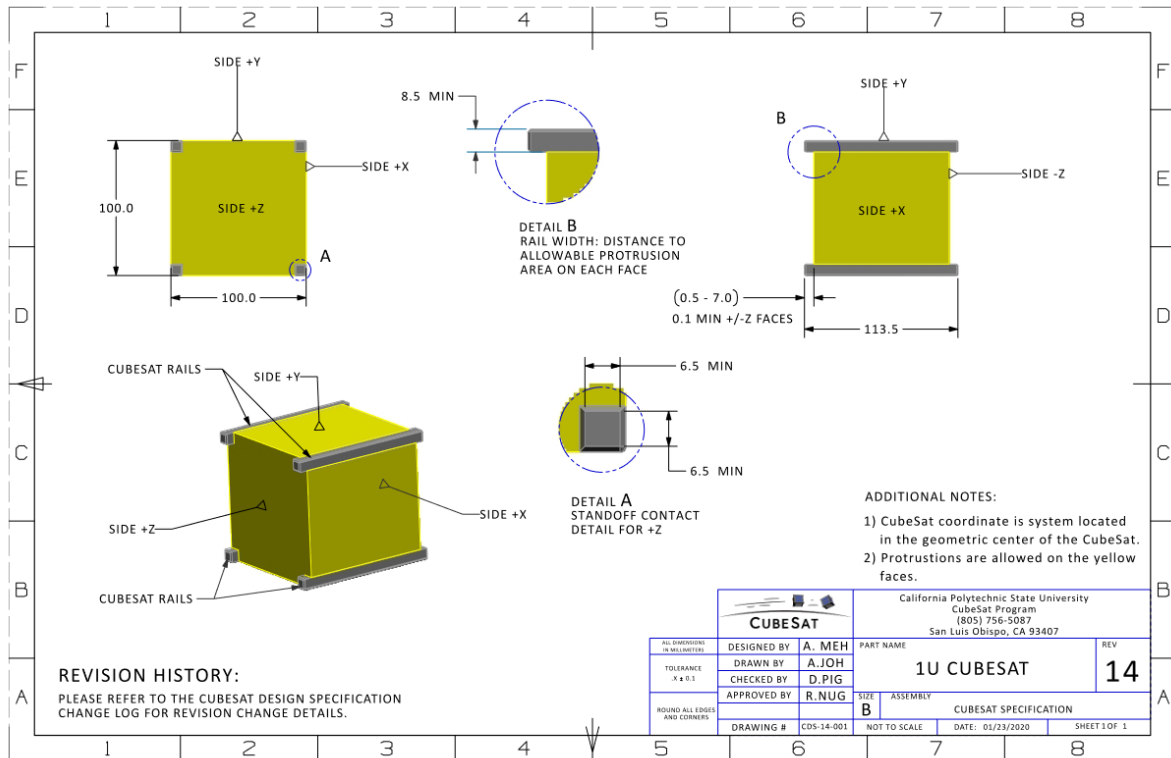


Рис. 5. Спецификация Cubesat 1U, предоставленная участникам

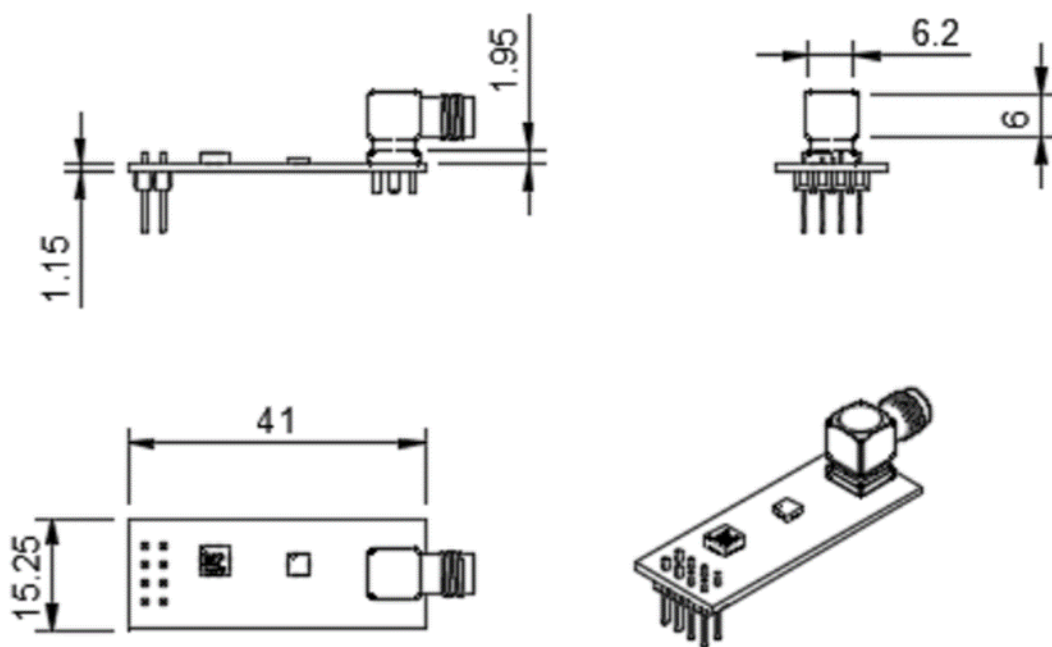


Рис. 6. Чертёж модуля NRF24L01 без антенны, предоставленный участникам

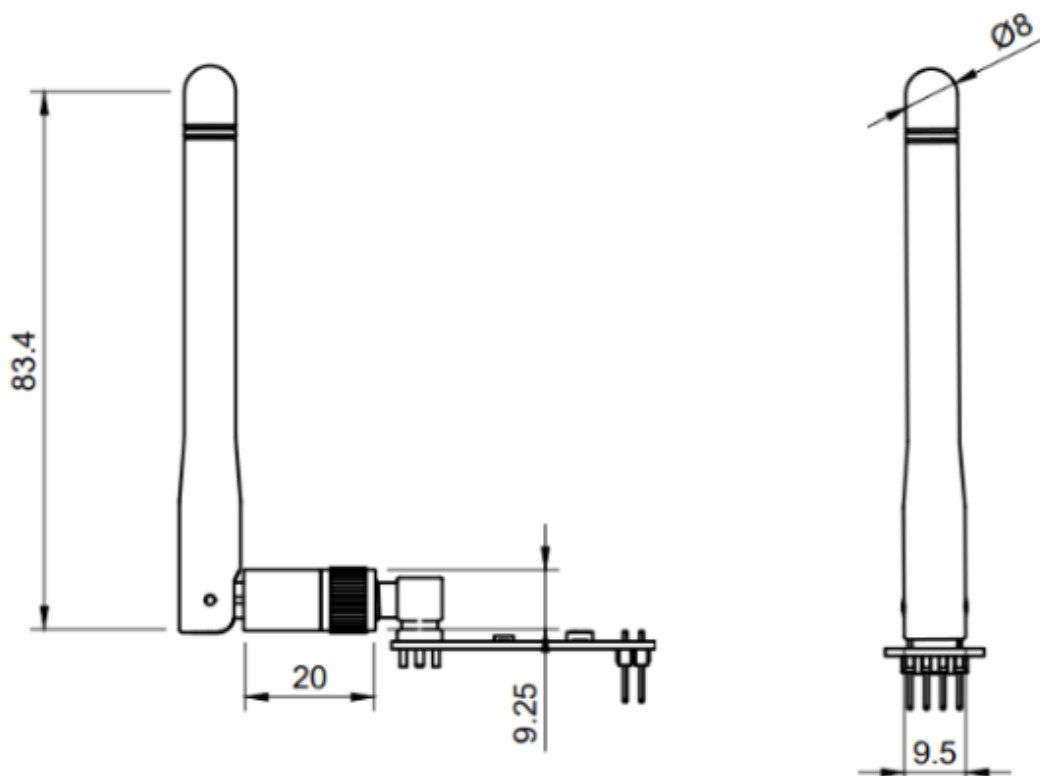


Рис. 7. Чертёж модуля NRF24L01 с антенной, предоставленный участникам

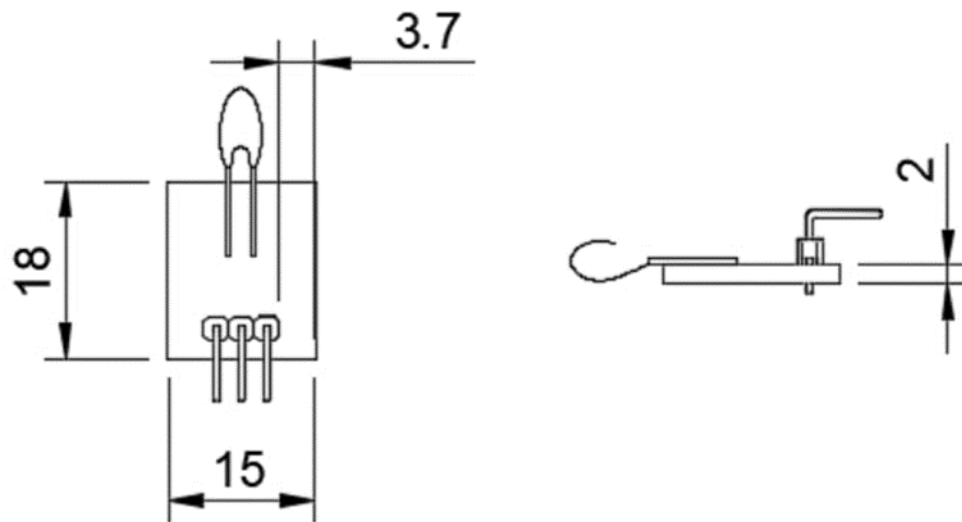


Рис. 8. Чертёж датчика температуры KY-013, предоставленный участникам

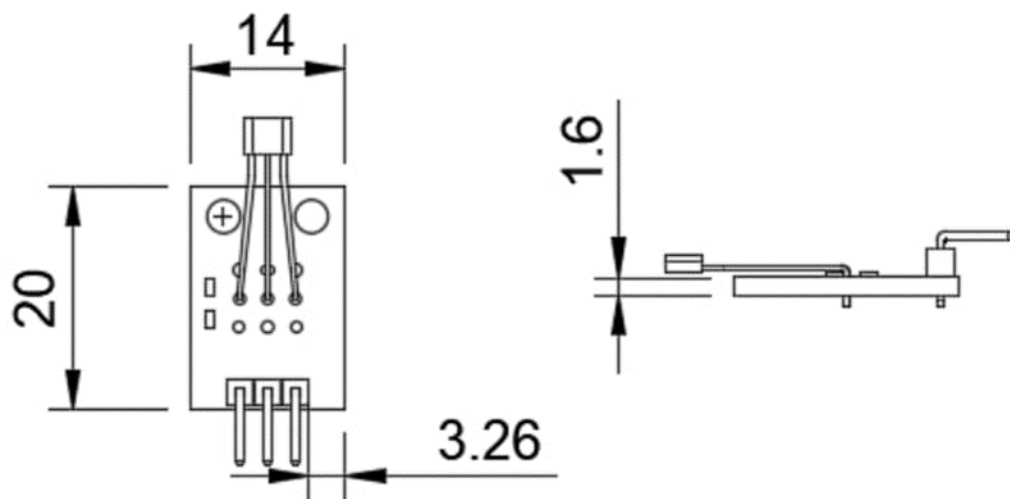


Рис. 9. Чертёж датчика магнитного поля KY-035, предоставленный участникам

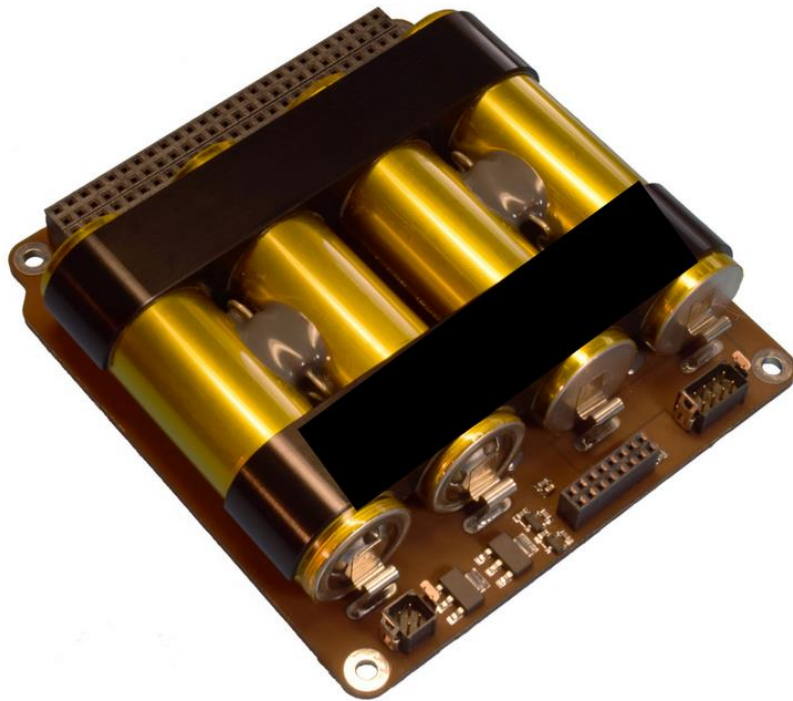


Рис. 10. Фото блока аккумуляторной батареи, предоставленное участникам

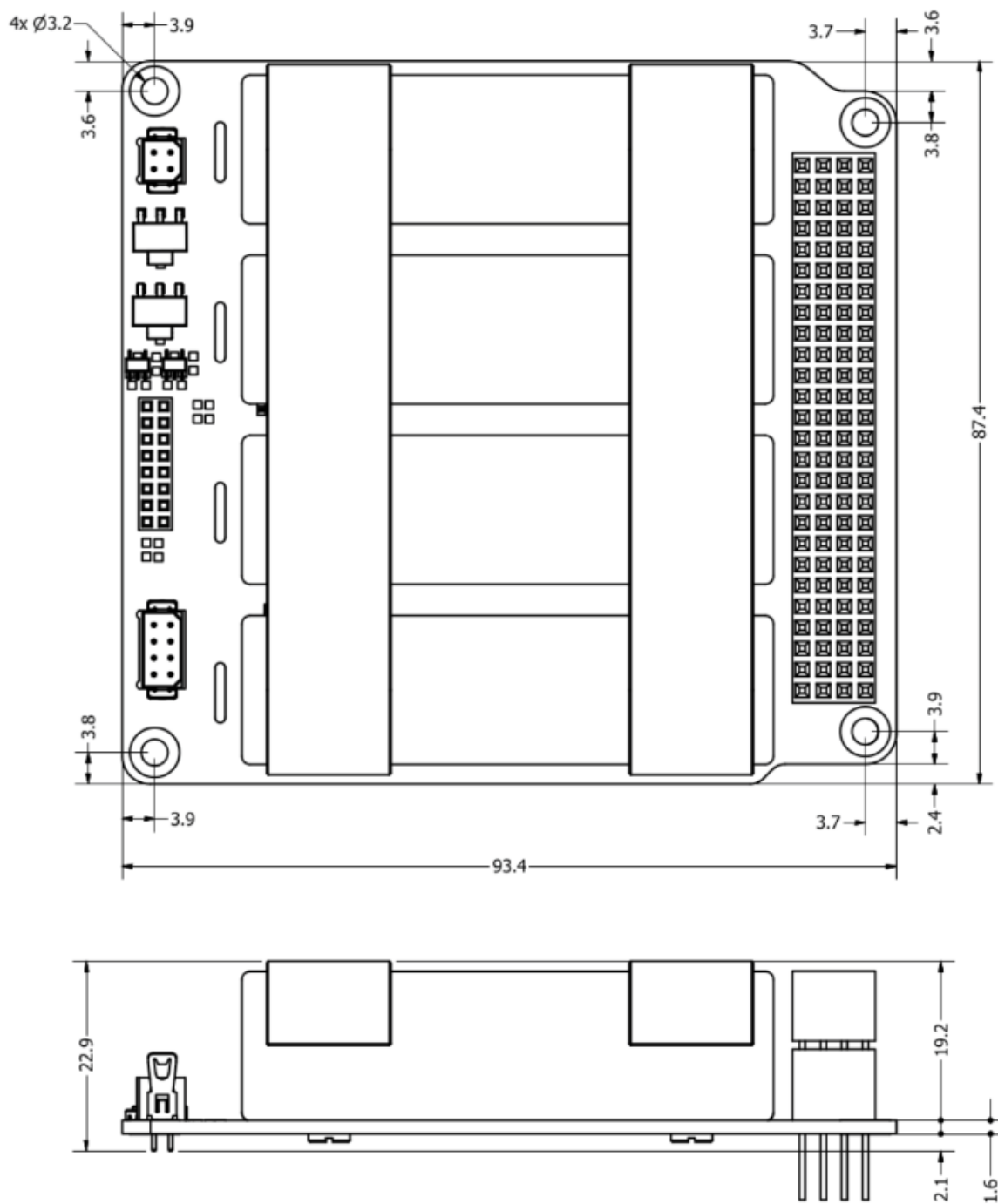


Рис. 11. Чертёж блока аккумуляторной батареи, предоставленный участникам

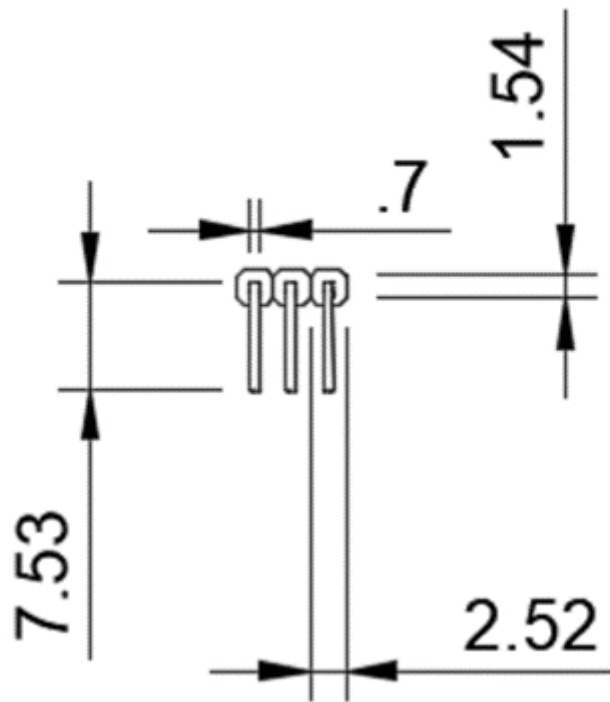


Рис. 12. Чертёж пинов, предоставленный участникам

Эксперт просит участника продемонстрировать на одном экране трехмерную модель и предоставленный чертеж. Эксперт просит участника показать поочередно виды сбоку, сверху и любые другие на усмотрение эксперта.

Баллы ставятся за:

- 3.1 Разработку из отдельных элементов.
- 3.2 Сходство с изначальным чертежом.
- 3.3 Соблюдение размеров.
- 3.4 Правильность разработки модели.
- 3.5 Соблюдение всех скруглений и правильность размещения отверстий.