

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

**Решение финальной задачи**

**практики «Инженерия»**

**Московской предпрофессиональной олимпиады.**

**«Операция «Контакт»»**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

**Этап №1. Сборка и монтаж устройств**

**Пункт 1.1 Сборка наземной лазерной станции**

Участник должен продемонстрировать собранную наземную лазерную станцию, состоящую из платы Arduino UNO с корректно подключёнными компонентами: лазерный модуль KY-008 на двухосевом поворотном креплении с двумя сервоприводами TS90D, радиомодуль NRF24L01, ЖК-дисплей MT-16S2H. Все компоненты размещаются на беспаячной макетной плате, установленной на настольной подложке.

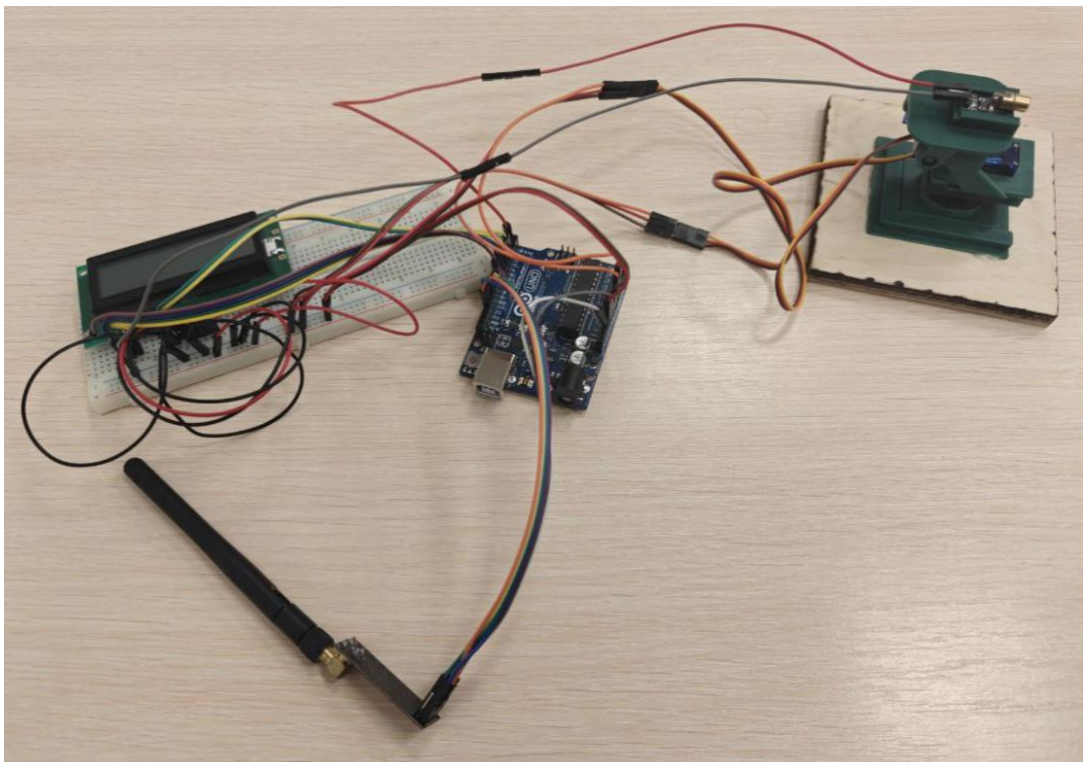


Рис. 1. Корректное подключение компонентов наземной станции

**Важно! Радиомодуль NRF24L01 должен быть подключён строго к 3.3 В. Подключение к 5 В выведет модуль из строя. Двухсторонний скотч на обратной стороне макетной платы не должен быть отклеен.**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

**Пункт 1.2 Сборка МКА в раме CubeSat 1U**

Участник должен продемонстрировать собранный малый космический аппарат (МКА) в раме CubeSat 1U, состоящий из платы Arduino UNO, солнечной панели 60×60 мм (используется как фотоприёмник лазерного сигнала), радиомодуля NRF24L01, датчика температуры KY-013 и датчика магнитного поля (датчик Холла) KY-035. Солнечная панель подключается к аналоговому входу Arduino.

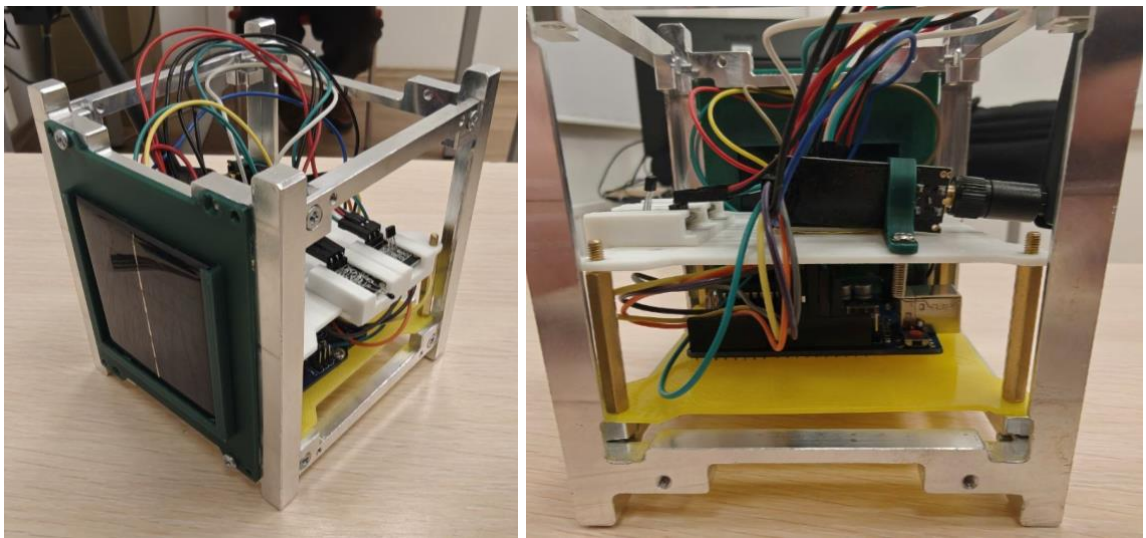


Рис. 2. Образец сборки МКА в раме CubeSat 1U

**Важно! NRF24L01 на МКА также подключается к 3.3 В. Солнечная панель подключается к аналоговому входу (A0–A5).**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

**Этап №2. Бортовая телеметрия**

**Пункт 2.1 Правильное подключение датчиков**

Участник должен продемонстрировать собранное устройство (МКА), к которому корректно подключены: датчик температуры КУ-013 (термистор, аналоговый вход), датчик магнитного поля КУ-035 (датчик Холла, аналоговый вход) и радиомодуль NRF24L01 для передачи телеметрии на наземную станцию.

**Пункт 2.2 Код для считывания и обработки данных с датчиков**

**Правильным считается наличие в программном коде МКА (подпункты 2.2а — 2.2d) следующих функций:**

- Считывание показаний датчика температуры КУ-013 и датчика магнитного поля КУ-035.
- Перевод значения температуры в градусы Цельсия с использованием формулы из Приложения 9.
- Передача телеметрических данных по радиоканалу NRF24L01 на наземную станцию (в открытом виде, без шифрования).
- Вывод телеметрии на ЖК-дисплей наземной станции в формате «Т:XX.X Н:XXXX» (вторая строка дисплея).

**Отсутствие требуемых фрагментов кода или орфографические ошибки считаются ошибками. Фрагменты кода могут иметь другой стиль написания, но описанный функционал должен выполняться.**

**Пункт 2.2а. Код считывания данных с датчиков**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

Участник демонстрирует программный код, осуществляющий считывание значений с подключённых датчиков и запись их в соответствующие переменные.

**Пример 1.**

```
int adc = analogRead(PIN_TEMP);    // A1 – термистор
KY-013
int hall = analogRead(PIN_HALL);   // A0 – датчик
Холла KY-035
```

**Пункт 2.2b. Код преобразования показаний термистора KY-013 в градусы Цельсия**

Участник реализует функцию, переводящую показания термистора, считанные с АЦП, в градусы Цельсия с использованием формулы из Приложения 9.

**Пример 1.**

```
float readTemperature() {
    int adc = analogRead(PIN_TEMP);
    if (adc <= 0) adc = 1;
    if (adc >= 1023) adc = 1022;
    double R = 10000.0 * adc / (1023.0 - adc);
    double T = 1.0 / (1.0/298.15 + (1.0/3950.0) *
log(R/10000.0));
    return (float) (T - 273.15);
}
```

**Пункт 2.2с. Код передачи телеметрии по радиоканалу NRF24L01**

Участник демонстрирует программный код, осуществляющий передачу телеметрических данных (температура и магнитное поле) по радиоканалу

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

NRF24L01 с МКА на наземную станцию. Участники самостоятельно определяют способ различения типов пакетов (телеметрия и зашифрованные координаты).

**Пример 1.**

```
void sendTelemetry() {
    float temp = readTemperature();
    int hall = analogRead(PIN_HALL);

    char packet[32];
    memset(packet, 0, 32);
    packet[0] = 'T';
    dtostrf(temp, 4, 1, packet + 1);
    int len = strlen(packet);
    packet[len] = ' ';
    sprintf(packet + len + 1, "%d", hall);

    radio.stopListening();
    radio.write(packet, 32);
    radio.startListening();
}
```

**Пункт 2.2d. Код вывода телеметрических данных на ЖК-дисплей наземной станции**

Участник демонстрирует программный код наземной станции, который принимает телеметрию по NRF24L01 и корректно выводит данные на вторую строку ЖК-дисплея в формате «Т:XX.X Н:XXXX», где Т — температура в градусах Цельсия (округлённая до одного знака), Н — значение магнитного поля (целое число, значение АЦП). Никаких лишних символов на экране выводиться не должно.

**Пример 1.**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

```
// Приём телеметрии на наземной станции:
if (packet[0] == 'T') {
    telemetryTemp = atof(packet + 1);
    char* sp = strchr(packet + 1, ' ');
    if (sp) telemetryHall = atoi(sp + 1);
    hasTelemetry = true;
}

// Вывод на дисплей (вторая строка):
lcd.setCursor(0, 1);
char line2[17];
char ts[7];
dtostrf(telemetryTemp, 4, 1, ts);
snprintf(line2, 17, "T:%s H:%-4d", ts,
telemetryHall);
lcd.print(line2);
```

### **Пункт 2.3 Работоспособность подсистемы телеметрии**

Проверяющий просит участника запустить оба устройства и продемонстрировать, что:

- Данные с датчиков точно считываются и передаются по радиоканалу на наземную станцию.
- На второй строке ЖК-дисплея наземной станции отображается телеметрия в формате «Т:XX.X Н:XXXX».
- Значение температуры соответствует комнатной с погрешностью не более  $\pm 5$  °С.
- При поднесении магнита к датчику КУ-035 значение Н на дисплее изменяется.
- Лишние символы на экране отсутствуют, радиопередача стабильна.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание

---

**Этап №3. Лазерная связь и шифрование**

**Пункт 3.1 Код наземной станции**

**Правильным считается наличие в программном коде наземной станции (подпункты 3.1а — 3.1е) следующих функций:**

- Автоматическое сканирование пространства лазерным лучом по двум осям (горизонтальная и вертикальная) с помощью двух сервоприводов.
- Обнаружение МКА: приём по NRF24L01 сигнала подтверждения захвата, немедленная остановка сканирования и фиксация сервоприводов.
- Передача ключа шифрования по лазерному каналу (синхронизация + побитовая передача по протоколу из задания).
- Приём зашифрованного сообщения от МКА по NRF24L01 и его дешифровка модифицированным шифром Виженера.
- Вывод расшифрованного сообщения на первую строку ЖК-дисплея.

**Отсутствие требуемых фрагментов кода или орфографические ошибки считаются ошибками. Фрагменты кода могут иметь другой стиль написания, но описанный функционал должен выполняться.**

**Пункт 3.1а. Код автоматического сканирования**

Участник демонстрирует программный код, реализующий автоматическое сканирование пространства лазерным лучом с использованием двух сервоприводов (горизонтальная и вертикальная оси). Паттерн сканирования, шаг и скорость участники определяют самостоятельно.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

**Пример 1.**

```
// Все паттерны сканирования определяются
участниками самостоятельно (ниже приведен один из
возможных вариантов сканирования)
const int H_MIN = 65, H_MAX = 115;
const int V_MIN = 35, V_MAX = 70;
const int SCAN_STEP = 2;

int posH = H_MIN, posV = V_MIN, dirH = 1;

void scanStep() {
    servoH.write(posH);
    servoV.write(posV);
    posH += dirH * SCAN_STEP;

    if (posH > H_MAX) {
        posH = H_MAX; dirH = -1;
        posV += SCAN_STEP;
        if (posV > V_MAX) posV = V_MIN;
    } else if (posH < H_MIN) {
        posH = H_MIN; dirH = 1;
        posV += SCAN_STEP;
        if (posV > V_MAX) posV = V_MIN;
    }
}
```

**Пункт 3.1b. Код обнаружения МКА и фиксации сервоприводов**

Участник демонстрирует программный код, реализующий приём сигнала подтверждения захвата от МКА по радиоканалу NRF24L01. При получении сигнала станция немедленно останавливает сканирование и фиксирует сервоприводы в текущем положении.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

**Пример 1.**

```
// В основном цикле:
if (radio.available()) {
    char packet[32] = {0};
    radio.read(packet, 32);

    if (packet[0] == 'A' && packet[1] == 'C' &&
packet[2] == 'K') {
        // МКА обнаружен – фиксируем сервоприводы
        currentState = STATE_LOCKED;
        delay(500); // стабилизация сервоприводов
        // Переход к передаче ключа
        currentState = STATE_TRANSMITTING;
    }
}
```

**Пункт 3.1с. Код передачи ключа шифрования по лазерному каналу**

Участник демонстрирует программный код, реализующий передачу ключа шифрования методом амплитудной модуляции лазера в соответствии с протоколом:

- Последовательность синхронизации: пауза 500 мс (LOW), преамбула 2000 мс (HIGH), пауза 500 мс (LOW).
- Побитовая передача: каждый символ — 8 бит по 100 мс каждый, MSB first. Бит «1» = лазер включён (HIGH), бит «0» = выключен (LOW).
- Межсимвольная пауза: 200 мс (LOW).
- Конец сообщения: 1000 мс (LOW).
- Циклический повтор с фазы синхронизации.

**Пример 1.**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

```
const char CIPHER_KEY[] = "VEKTOR";
const unsigned long BIT_DURATION = 100;
const unsigned long CHAR_GAP = 200;
const unsigned long SYNC_LOW = 500;
const unsigned long PREAMBLE_HIGH = 2000;
const unsigned long END_LOW = 1000;

void transmitByte(char ch) {
    for (int bit = 7; bit >= 0; bit--) {
        digitalWrite(PIN_LASER, (ch & (1 << bit)) ?
HIGH : LOW);
        delay(BIT_DURATION);
    }
}

void transmitKey() {
    digitalWrite(PIN_LASER, LOW);    delay(SYNC_LOW);
    digitalWrite(PIN_LASER, HIGH);
    delay(PREAMBLE_HIGH);
    digitalWrite(PIN_LASER, LOW);    delay(SYNC_LOW);

    for (int i = 0; CIPHER_KEY[i] != '\0'; i++) {
        transmitByte(CIPHER_KEY[i]);
        if (CIPHER_KEY[i + 1] != '\0') {
            digitalWrite(PIN_LASER, LOW);
            delay(CHAR_GAP);
        }
    }
    digitalWrite(PIN_LASER, LOW);
    delay(END_LOW);
}
```

**Пункт 3.1d. Код приёма зашифрованного сообщения и дешифровки**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

Участник демонстрирует программный код наземной станции, принимающий зашифрованное сообщение от МКА по NRF24L01 и дешифрующий его модифицированным шифром Виженера (описание метода — Приложение 2).

**Пример 1.**

```
int getShift(char keyChar) {
    return (keyChar - 'A') % 10;
}

void decryptVigenere(const char* cipher, const
char* key, char* out) {
    int keyLen = strlen(key);
    int i;
    for (i = 0; cipher[i] != '\0'; i++) {
        char k = key[i % keyLen];
        char ch = cipher[i];
        if (ch >= 'A' && ch <= 'Z') {
            out[i] = 'A' + ((ch - 'A' - (k - 'A') + 26) %
26);
        } else if (ch >= '0' && ch <= '9') {
            out[i] = '0' + ((ch - '0' - getShift(k) + 10)
% 10);
        } else {
            out[i] = ch;
        }
    }
    out[i] = '\0';
}

// При приёме пакета с префиксом 'C':
if (packet[0] == 'C') {
    decryptVigenere(packet + 1, CIPHER_KEY,
decryptedMsg);
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

```
hasCoordinates = true;  
}
```

### **Пункт 3.1е. Код вывода результата на ЖК-дисплей**

Участник демонстрирует программный код, выводящий расшифрованное сообщение (координаты) на первую строку ЖК-дисплея. До получения сообщения на первой строке выводится текущий статус работы станции (например, «SCANNING», «LOCK», «RECEIVING»). Лишних символов на экране быть не должно.

#### **Пример 1.**

```
// Вывод статуса до получения координат:  
void setStatus(const char* s) {  
    if (!hasCoordinates) {  
        lcd.setCursor(0, 0);  
        lcd.print(s);  
        for (int i = strlen(s); i < 16; i++)  
            lcd.print(' ');  
    }  
}  
  
// Вывод расшифрованного сообщения:  
lcd.setCursor(0, 0);  
lcd.print(decryptedMsg);  
for (int i = strlen(decryptedMsg); i < 16; i++)  
    lcd.print(' ');
```

### **Пункт 3.2 Код МКА**

**Правильным считается наличие в программном коде МКА (подпункты 3.2а — 3.2е) следующих функций:**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

- Калибровка: серия замеров АЦП для определения базового уровня фоновой освещённости (50–100 замеров за 1–2 секунды). Определение порога срабатывания.
- Обнаружение лазерного луча: при превышении порога — отправка сигнала подтверждения захвата по NRF24L01.
- Приём ключа шифрования по лазерному каналу: ожидание последовательности синхронизации, побитовый приём данных (100 мс/бит, MSB first).
- Шифрование координат (секретное сообщение из Приложения 4) модифицированным шифром Виженера с использованием принятого ключа.
- Передача зашифрованного сообщения по NRF24L01 на наземную станцию.

**Пункт 3.2а. Код калибровки и определения порога**

Участник демонстрирует программный код, реализующий калибровку при включении МКА: серия замеров значений АЦП солнечной панели (рекомендуется 50–100 замеров за 1–2 секунды) для определения базового уровня фоновой освещённости. Порог срабатывания определяется как: порог = базовый уровень +  $\Delta$ , где  $\Delta$  подбирается экспериментально.

**Пример 1.**

```
const int DELTA = 20;
int baseLevel = 0;

// Калибровка при запуске (setup):
long sum = 0;
for (int i = 0; i < 100; i++) {
    sum += analogRead(PIN_SOLAR);
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

```
    delay(10);  
}  
baseLevel = sum / 100;
```

**Пункт 3.2в. Код обнаружения лазера и отправки подтверждения захвата**

Участник демонстрирует программный код, реализующий обнаружение попадания лазерного луча на солнечную панель (превышение порога АЦП) и отправку сигнала подтверждения захвата по радиоканалу NRF24L01 на наземную станцию.

**Пример 1.**

```
bool laserHigh() {  
    int sum = 0;  
    for (int i = 0; i < 5; i++) {  
        sum += analogRead(PIN_SOLAR);  
    }  
    return (sum / 5) > (baseLevel + DELTA);  
}
```

// При обнаружении лазера – подтверждение 3 из 5 замеров:

```
if (laserHigh()) {  
    int confirm = 0;  
    for (int i = 0; i < 5; i++) {  
        if (laserHigh()) confirm++;  
        delay(3);  
    }  
    if (confirm >= 3) {  
        char ack[32];  
        memset(ack, 0, 32);  
        strcpy(ack, "ACK");  
        radio.stopListening();  
    }  
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

```
for (int i = 0; i < 5; i++) {  
    radio.write(ack, 32);  
    delay(5);  
}  
radio.startListening();  
}  
}
```

**Пункт 3.2с. Код приёма ключа по лазерному каналу**

Участник демонстрирует программный код, реализующий приём ключа шифрования по лазерному каналу в соответствии с протоколом передачи данных:

- Ожидание последовательности синхронизации (пауза 500 мс, преамбула 2000 мс, пауза 500 мс).
- Побитовый приём данных: чтение состояния АЦП каждые 100 мс, формирование байтов (MSB first, 8 бит на символ).
- Межсимвольная пауза 200 мс.
- Определение конца сообщения (пауза 1000 мс).

**Пример 1.**

```
bool waitForSync(unsigned long timeoutMs, unsigned  
long &syncEndTime) {  
    unsigned long start = millis();  
    while (millis() - start < timeoutMs) {  
        while (!laserHigh()) {  
            if (millis() - start > timeoutMs) return  
false;  
        }  
        unsigned long highStart = millis();  
        while (laserHigh()) {
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

```
        if (millis() - start > timeoutMs) return
false;
    }
    unsigned long highDur = millis() - highStart;
    if (highDur >= 1500) {
        delay(500);
        syncEndTime = millis();
        return true;
    }
}
return false;
}

bool receiveKey() {
    int keyIndex = 0;
    for (int charIdx = 0; charIdx < 30; charIdx++) {
        uint8_t val = 0;
        unsigned long charStart = rxSyncEnd + (unsigned
long)charIdx * 1000UL;
        for (int bit = 7; bit >= 0; bit--) {
            unsigned long sampleTime = charStart +
(unsigned long)(7 - bit) * 100UL + 50UL;
            while (millis() < sampleTime) {}
            if (laserHigh()) val |= (1 << bit);
        }
        if (val >= 'A' && val <= 'Z') {
            receivedKey[keyIndex++] = (char)val;
            receivedKey[keyIndex] = '\0';
        } else {
            break;
        }
    }
    return keyIndex > 0;
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

**Пункт 3.2d. Код шифрования координат модифицированным шифром Виженера**

Участник реализует функцию шифрования секретного сообщения (координаты центра управления ВЕКТОРа из Приложения 4) модифицированным шифром Виженера с использованием ключа, принятого по лазерному каналу (описание метода — Приложение 2).

**Пример 1.**

```
int getShift(char keyChar) {
    return (keyChar - 'A') % 10;
}

void encryptVigenere(const char* plain, const char*
key, char* out) {
    int keyLen = strlen(key);
    int i;
    for (i = 0; plain[i] != '\0'; i++) {
        char k = key[i % keyLen];
        char ch = plain[i];
        if (ch >= 'A' && ch <= 'Z') {
            out[i] = 'A' + ((ch - 'A' + (k - 'A')) % 26);
        } else if (ch >= '0' && ch <= '9') {
            out[i] = '0' + ((ch - '0' + getShift(k)) %
10);
        } else {
            out[i] = ch;
        }
    }
    out[i] = '\0';
}
```

**Пункт 3.2e. Код передачи зашифрованного сообщения по NRF24L01**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Практика “Инженерия”**  
**Финальное практическое задание**

---

Участник демонстрирует программный код, осуществляющий передачу зашифрованного сообщения по радиоканалу NRF24L01 на наземную станцию.

**Пример 1.**

```
void sendEncrypted(const char* msg) {
    char packet[32];
    memset(packet, 0, 32);
    packet[0] = 'C';
    strncpy(packet + 1, msg, 30);

    radio.stopListening();
    radio.write(packet, 32);
    radio.startListening();
}
```

**Пункт 3.3 Качество работы системы лазерной связи**

Проверяющий просит участников разместить наземную станцию у линии красного скотча. Жюри устанавливает МКА на линии зелёного скотча (70 см от станции) в произвольном горизонтальном положении. МКА может быть установлен на столе (0 см) или на фанерной подставке (7–10 см) — вариант определяется жюри.

По команде жюри участники запускают оба устройства одновременно. Проверяющий наблюдает следующую последовательность:

- Наземная станция автоматически сканирует пространство лазерным лучом.
- При обнаружении МКА сервоприводы фиксируются, начинается передача ключа по лазеру.
- МКА принимает ключ, шифрует координаты и передаёт зашифрованное сообщение по радиоканалу.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание

---

- Наземная станция принимает, дешифрует и выводит результат на первую строку дисплея.
- Результат на дисплее должен совпадать с эталонным значением из Приложения 4.

Предоставляется до 3 попыток; в зачёт идёт лучший результат. Между попытками жюри может изменить положение МКА.

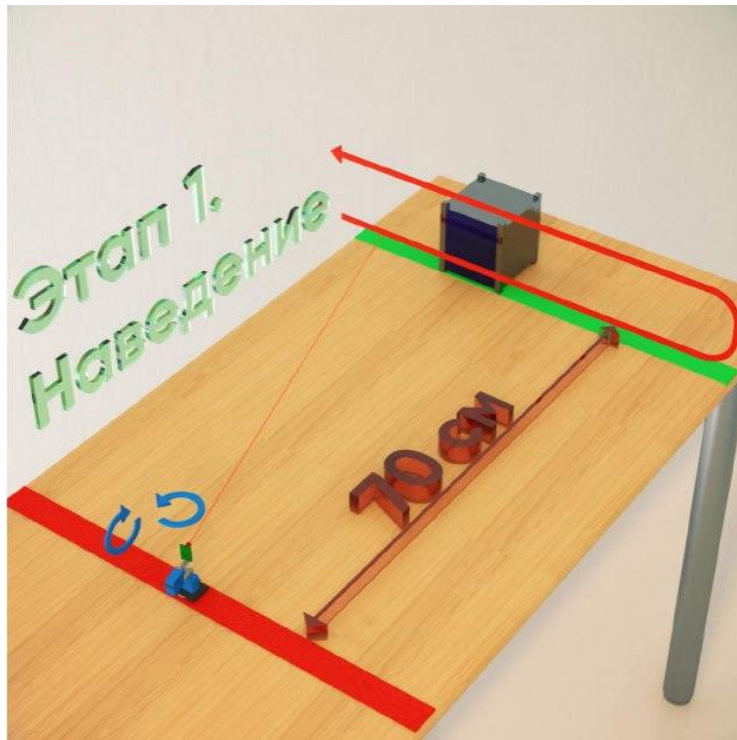


Рис. 5. Схема рабочего места для проведения испытания 1 (наведение)

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание

---

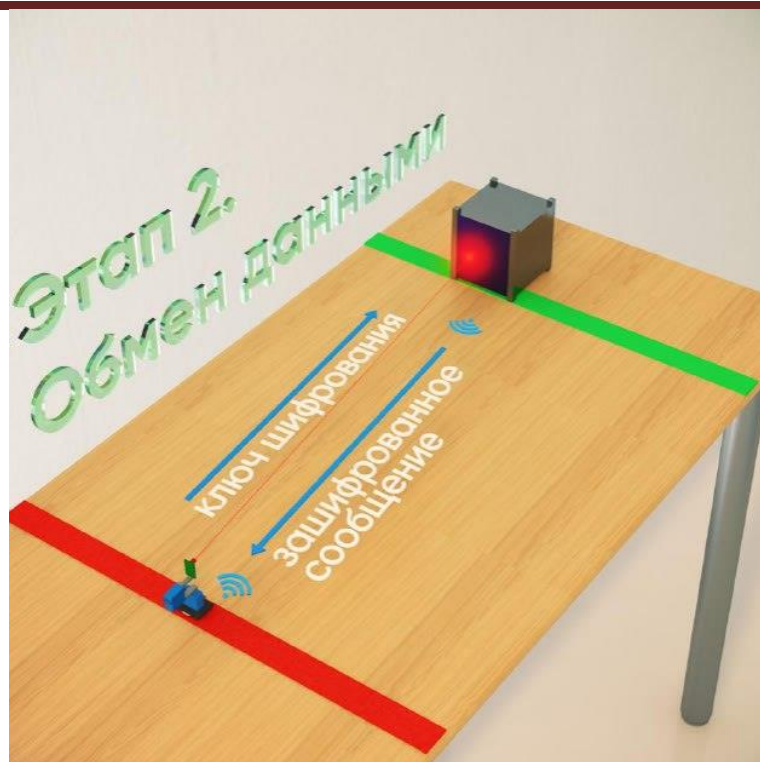
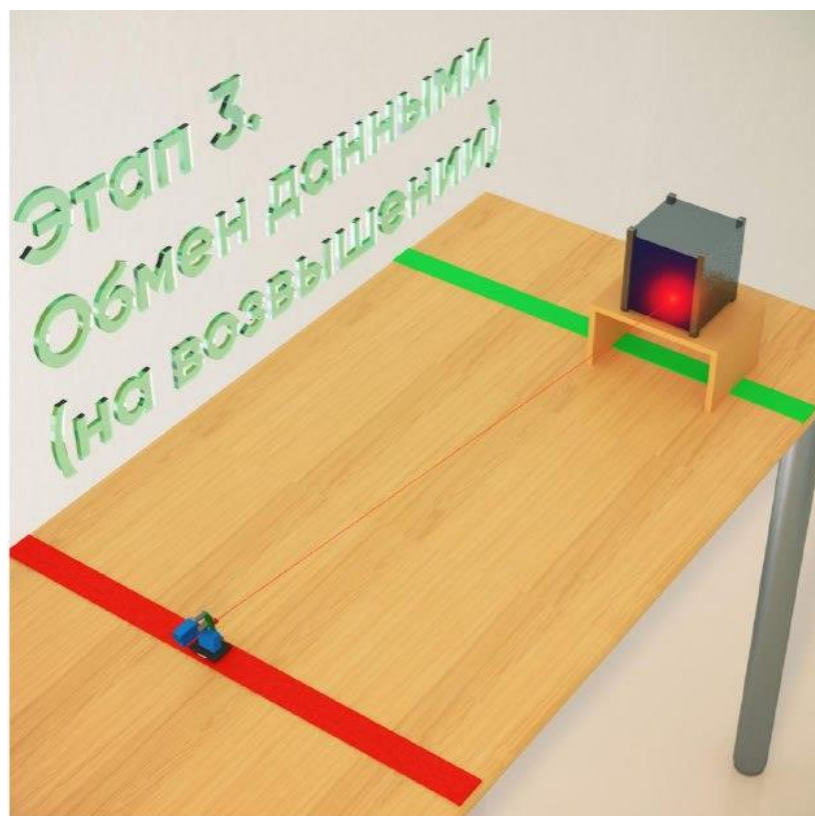


Рис. 6. Схема рабочего места для проведения испытания 2 (обмен данными)



**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

Рис. 7. Схема рабочего места для проведения испытания 3 (обмен данными на подставке)

**Таблица 1. Образец результата (расшифрованное сообщение на дисплее)**

Параметр	Значение
Ключ шифрования (Приложение 4)	VEKTOR
Секретное сообщение (Приложение 4)	N55E037ALT400
Зашифрованный результат	I95X408EVM871
Расшифрованный результат на дисплее	N55E037ALT400

### **Резервные варианты**

- Если не реализовано автоматическое сканирование — участники направляют лазер вручную. Баллы за автосканирование не начисляются.
- Если не реализован приём ключа по лазерному каналу — ключ из Приложения 4 прошивается в программу МКА заранее. Баллы за лазерный приём не начисляются.
- Если не реализовано программное шифрование — участники выполняют шифрование вручную на бумаге. Баллы за программное шифрование не начисляются, но начисляются частичные баллы за понимание метода.
- Если не реализована обратная радиопередача — зашифрованное сообщение демонстрируется в Serial Monitor. Баллы за радиопередачу обратно не начисляются.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

**Этап №4. 3D-моделирование корпуса МКА формата CubeSat 1U**

Участник должен продемонстрировать трёхмерную модель рамы CubeSat 1U, трёхмерные модели всех компонентов (датчики, солнечные панели, блок АКБ, крепления) и 3D-сборку итогового МКА, разработанные в соответствии с чертежами из Приложений 10 и 11.

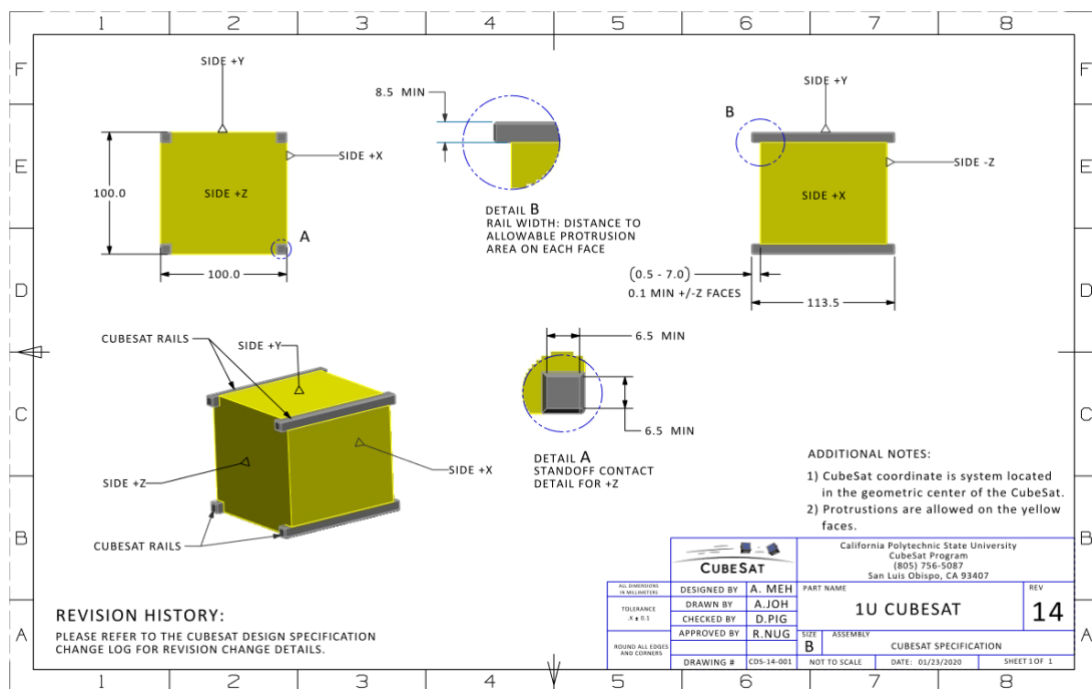


Рис. 8. Чертёж рамы CubeSat 1U, предоставленный участникам (Приложение 10)

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание

---

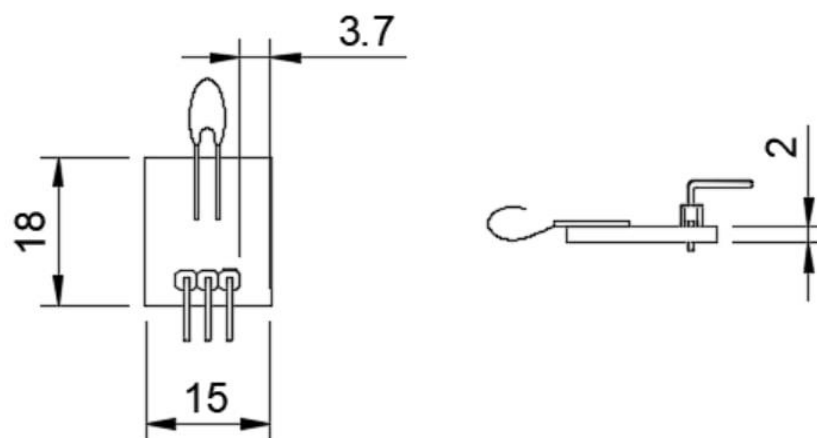


Рис. 10. Чертёж датчика температуры KY-013, предоставленный участникам

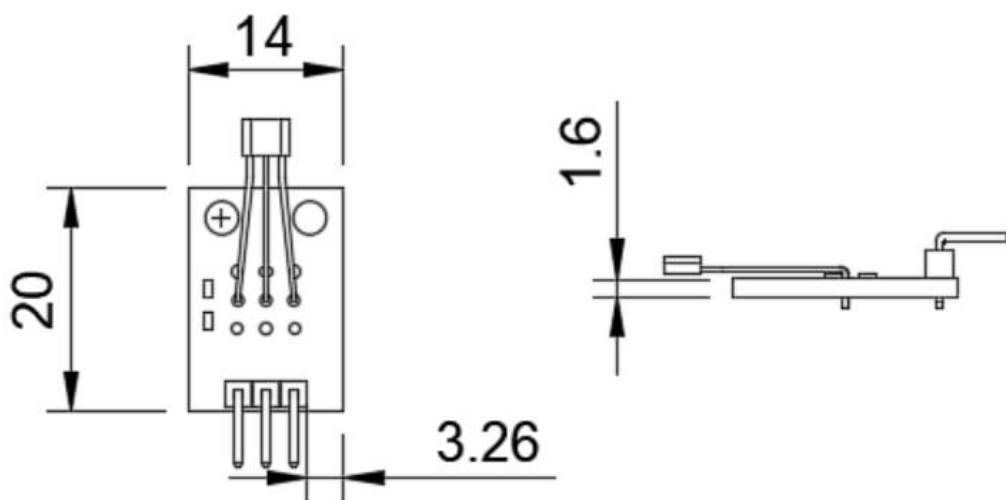


Рис. 11. Чертёж датчика магнитного поля KY-035, предоставленный участникам

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание

---

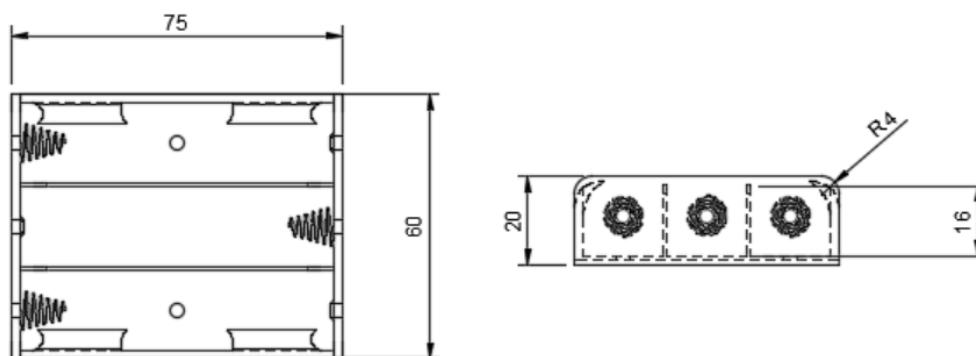


Рис. 12. Чертёж держателя аккумуляторов, предоставленный участникам  
(Приложение 11)

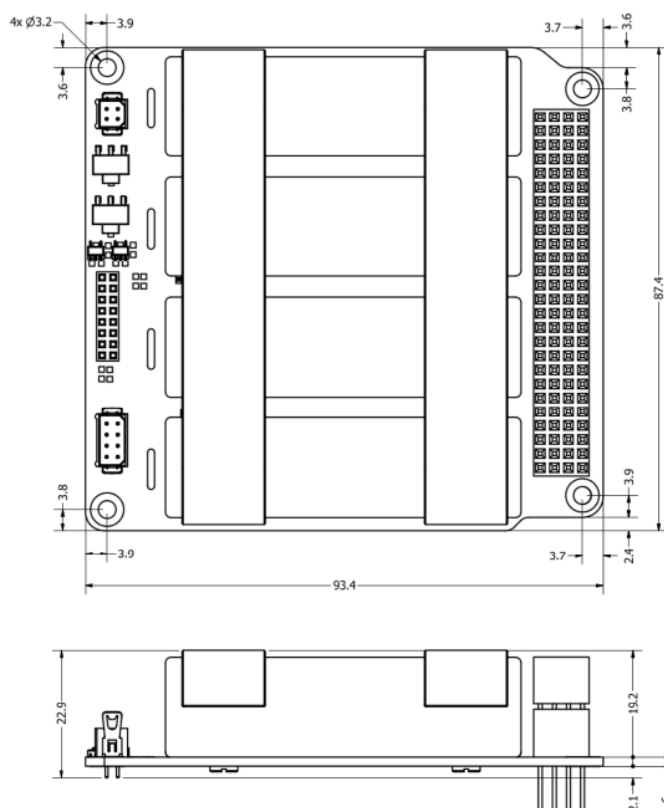


Рис. 13. Чертёж блока аккумуляторной батареи, предоставленный  
участникам (Приложение 11)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Практика “Инженерия”  
Финальное практическое задание**

---

Эксперт просит участника продемонстрировать трёхмерную модель, предоставленные чертежи и 3D-сборку всего космического аппарата. Эксперт просит участника показать поочерёдно виды сбоку, сверху и любые другие на усмотрение эксперта.

**Баллы ставятся за:**

- 4.1 Разработку из отдельных элементов.
- 4.2 Сходство с изначальным чертежом.
- 4.3 Соблюдение размеров.
- 4.4 Правильность разработки модели.
- 4.5 Корректность сборки и взаимного расположения элементов (отсутствие пересечений тел).