



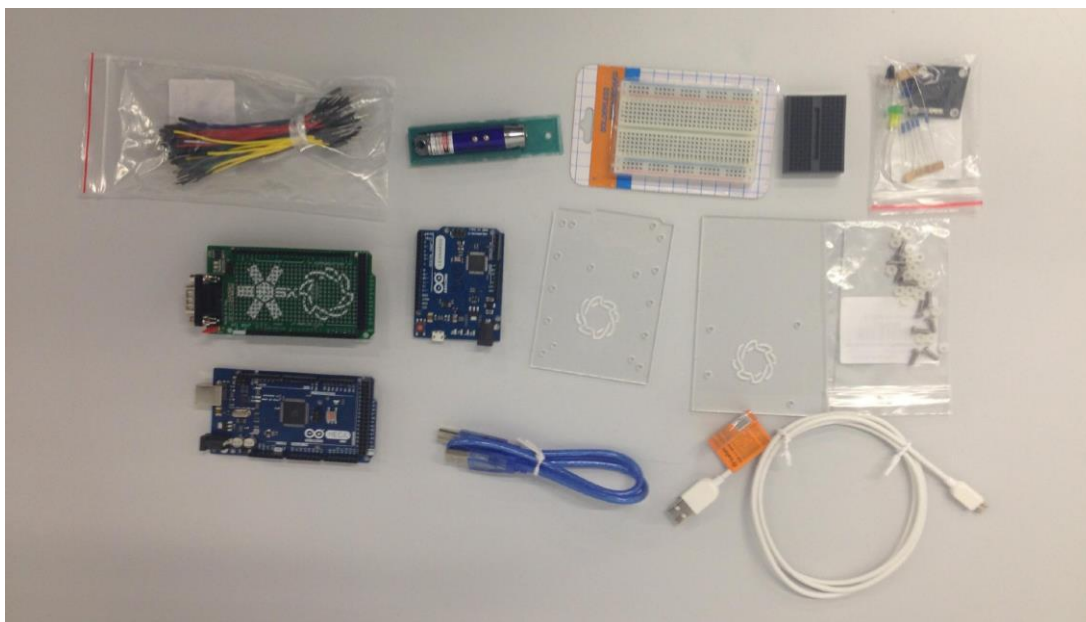
## §4 Заключительный этап: командная часть

Задания финального этапа предлагали наличие 2 независимых линий разработки, которые в итоге сольются в одно финальное изделие. Задания давались последовательно по дням, от переосмысления тем хактонов и заочных заданий к новым этапам выполняемым на полученном оборудовании. Баллы за задания росли по мере появления новых вводных составляющих. Команды должны были разделиться по темам проработки заданий. Одна часть команды должна реализовать ориентацию и стабилизацию вращения спутника, в то время как вторая реализовать программный код модуля связи. После этого происходит интеграция нового модуля в существующую инфраструктуру спутника.

Используемые материалы и оборудование:



Набор для сборки спутника: система электропитания, бортовой компьютер, модуль маховика, 4 солнечных датчика, датчик угловой скорости, укв приемник, корпусные детали, набор крепежных винтов, шлейф кабельной сети, разъёмы, обжимной инструмент, тестер шлейфа, отвертка, адаптер питания.



Набор для сборки модуля связи: плата arduino mega, плата arduino leonardo, шилд адаптера бортовой сети спутника, две беспаячные макетные платы, набор проводов, Модуль с ик светодиодом, ик фотоприемник, набор резисторов и светодиодов для отладки, лазерная указка с креплением, монтажные пластины из оргстекла и набор винтов, кабели USB для программирования плат.



Общее оборудование: Два ноутбука, настольная лампа, дополнительный адаптер usb-wifi. Кроме этого каждое место было оборудовано нитью с карабином для тестирования спутника и розетками 220 вольт для зарядки ноутбуков, подключения зарядки спутника и тестовой лампы.

Общая тестовая зона содержала 4 стенда из прожектора, рамки для подвеса спутника и тестовой мишени в расчете угла отклонения 5 градусов, стенда планшетсат терра и программного комплекса ЦУП.

## 4.1 Конкурсные задания

#### **Задание 4.1.1. Подготовка к работе**

*Условие:*

Используя инструкции по сборке и запуску спутника, необходимо собрать спутник и продемонстрировать его работу.

*Используемое оборудование:*

Конструктор для сборки модели спутника, сборочный инструмент, ноутбук.

*Критерий оценки:*

- Сборка спутника не противоречит его работе (каркас собран, расположение датчиков обеспечивает их нормальную работу, кабельная сеть не мешает работе датчиков) - **1 балл**
- На конструкторе запускается, и корректно работает тестовая программа - **1 балл**

*Комментарий:*

Для выполнения задания необходимо собрать конструкцию, по инструкциям с сайта [wiki.orbcraft.ru](http://wiki.orbcraft.ru) обеспечивающую штатное функционирование всех модулей спутника. Для этого необходимо на своем рабочем месте собрать каркас спутника, смонтировать кабельную сеть и запустить примеры из описания.

#### **Задание 4.1.2. Ориентации и стабилизации движения**

*Условие:*

Одной из ключевых задач, обеспечивающих успех научной или прикладной миссии выполняемой космическим аппаратом является задача ориентации и стабилизации движения. Успешное её решение сказывается на функционировании систем электропитания и связи, а также работе полезной нагрузки аппарата. В современной практике используются различные системы ориентации, которые как правило сочетаются друг с другом. Мы останавливаем внимание на одной из основных систем, которая позволяет быстро менять ориентацию спутника и обеспечивать ее поддержание в течение длительного времени. Такие системы ориентации применяются в большинстве современных космических аппаратов.

Вам предлагается реализовать алгоритм, который эффективно стабилизирует спутник в заданном положении.

*Используемое оборудование:*

Конструктор для сборки модели спутника, лазерная указка, стенд с прожектором, ноутбук.

*Критерий оценки:*

- Начальное положение и вращение спутника неизвестно (задается жюри). Спутник ориентируется в сторону мишени, после чего луч лазерной указки остается на мишени.
- Оценивается время непрерывного нахождения лазерного луча на мишени, начиная с первого пересечения мишени, из расчета 10 секунд – **1 балл**, но не более **10 баллов**.
- Общее время с момента запуска программы не более 360 секунд.

*Комментарий:*

Для выполнения задания нужно изучить примеры кода изложенные в руководстве <http://nti-contest.ru/wp-content/uploads/7-%D0%9A%D0%A1.pdf> и реализовать код

описанный в заданиях прошлого года. Это задание необходимо для успешного выполнения задания по передаче данных со спутника.

### **Задание 4.1.3. Оптический канал связи**

*Условие:*

Канал связи это система технических средств и среда распространения сигналов для односторонней передачи данных (информации) от отправителя (источника) к получателю (приемнику). Канал связи является составной частью канала передачи данных.

Для реализации канала связи необходимо обеспечить совместимость приемника и передатчика на физическом и логическом уровнях.

Физический уровень описывает способы передачи бит через физические среды линий связи, соединяющие устройства. На этом уровне описываются параметры сигналов, такие как амплитуда и частота, используемая модуляция и другие низкоуровневые параметры.

Протокол передачи данных — набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами. Эти соглашения задают единообразный способ передачи сообщений и обработки ошибок при взаимодействии программного обеспечения разнесенной в пространстве аппаратуры, соединенной тем или иным каналом связи.

*Используемое оборудование:*

Платы arduino mega (с передающей стороны), Arduino Leonardo (с принимающей стороны) Беспаяная макетная плата - 2 шт. Модуль с ИК-светодиодом, ИК-приёмник TSOP4856, набор проводов для макетной платы, ноутбук 2 шт

*Критерии оценки:*

- Данные должны передаваться: С компьютера №1 на Arduino №1 через виртуальный COM-порт; С Arduino №1 модулированным оптическим сигналом на Arduino №2; С Arduino №2 на компьютер №2 через виртуальный COM-порт.
- В результате выполнения задания необходимо составить работоспособную принципиальную электрическую схему **1 балл**, согласно схеме выполнить сборку схемы на макетной плате **1 балл**, и продемонстрировать возможность передачи данных с помощью собранного устройства – **1 балл**.
- Канал связи должен верно передать произвольные битовые массивы, длиной до 100 символов (выдаются судьей при сдаче задачи). Оценивается количество верно переданных массивов (всего 10, **1 балл** за каждый верно переданный массив).
- Код подсказки можно получить потеряв 50 процентов баллов за задание.

*Комментарий:*

Это задание является логичным следствием хакатона, материалы которого доступны участникам по ссылке <http://nti-contest.ru/wp-content/uploads/2016/09/%D0%9F%D0%BB%D0%B0%D0%BD-%D1%85%D0%B0%D0%BA%D0%B0%D1%82%D0%BE%D0%BD%D0%B0%E2%84%961.->

<http://nti-contest.ru/wp-content/uploads/2016/09/%D0%9A%D0%BE%D1%81%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B.pdf>

Код подсказки реализует передачу последовательности байт 0x00 – 0xff и прием их с преобразованием в десятичный формат. Для решения задачи нужно заменить в коде передачи отправляемый байт на передаваемую информацию а в коде приемника изменить формат представления на шестнадцатеричный.

*Код подсказки:*

**Отправляющая часть**

```
#include <stdint.h>

const uint8_t DIODE_PIN = 11;

const unsigned long UART_BAUD_RATE = 600;
const unsigned long UART_DELAY_US = 1000000 / UART_BAUD_RATE;

const unsigned long CARRIER_FREQUENCY = 56000;
const unsigned long CARRIER_DELAY_CYCLES = F_CPU / (CARRIER_FREQUENCY
* 2);

static void initPWM()
{
    TCCR1A = 0;
    TCCR1B = 0;
    TCCR1C = 0;
    TIMSK1 = 0;
}

static void startPWM()
{
    TCCR1B = 0;
    TCCR1A = 0;
    TCNT1 = 0;
    OCR1A = CARRIER_DELAY_CYCLES - 1;
    TCCR1A = (0 << COM1A1) | (1 << COM1A0) |
            (1 << WGM11) | (1 << WGM10);
    TCCR1B = (1 << WGM13) | (1 << WGM12) |
            (0 << CS12) | (0 << CS11) | (1 << CS10);
}

static void stopPWM()
{
    TCCR1B = 0;
    TCCR1A = 0;
}

static void sendZero()
{
    startPWM();
    delayMicroseconds(UART_DELAY_US);
}

static void sendOne()
{
    stopPWM();
    delayMicroseconds(UART_DELAY_US);
}

void setup()
{
    pinMode(DIODE_PIN, OUTPUT);
    digitalWrite(DIODE_PIN, LOW);
}
```

```

    initPWM();
}

void sendByte(uint8_t byte)
{
    sendZero();

    for (int i = 0; i < 8; i++) {
        Serial.println(byte & 1);
        if ((byte & 1) != 0) {
            sendOne();
        } else {
            sendZero();
        }

        byte >>= 1;
    }

    sendOne();
}

void loop()
{
    for (int i = 0; i < 256; i++) {
        sendByte(i);
        //delay(1000);
    }
}

```

### Приемная часть

```

#include <stdint.h>

void setup()
{
    Serial.begin(115200);
    Serial1.begin(600);
}

void loop()
{
    if (Serial1.available() > 0) {
        uint8_t received_byte = Serial1.read();
        Serial.println(String(received_byte));
    }
}

```

#### **Задание 4.1.4. Неизвестная планета**

*Условие:*

Неизвестная планета, очень похожая на Землю, испускает в укв радиодиапазоне сигналы. Ученые для изучения этой планеты запускают к ней космический аппарат. Ваша задача проверить гипотезу и попытаться понять, как сигналы зависят от смены дня и ночи на планете.

*Используемое оборудование:*

Конструктор для сборки модели спутника, модель планеты, ноутбук.

*Критерии оценки:*

- Сигнал от планеты нужно увидеть в консоли и идентифицировать. Если Вам удастся принять данные, которые планета вам посылает, вы получите **5 баллов**.
- Если Вы сможете определить зависимость сигналов от долготы, над которой пролетает ваш спутник, то получите еще **3 балла**

*Комментарий:*

Для выполнения задания нужно применить пример кода с сайта [wiki.orbcraft.ru](http://wiki.orbcraft.ru) найдя закономерность в принимаемых сообщениях

*Решение:*

```
#include "libschat.h"
/* ** Lab 9: UHF transceiver demo. */
void control(void) {
    const uint16_t tx_num = 2;
    const uint16_t rx_num = 1;
    const char hello[] = "hello, world!";
    printf("Enable transceiver #%d\n", tx_num);
    transceiver_turn_on(tx_num);
    Sleep(1);
    bus_setup();
    printf("Send data from #%d to #%d\n", tx_num, rx_num);
    if (LSS_OK != transceiver_send(tx_num, rx_num, (uint8_t *)
        hello, sizeof(hello)))
        puts("Fail!");
    printf("received #%d\n", rx_num);

    printf("Disable transceiver #%d\n", tx_num);
    transceiver_turn_off(tx_num);
    return;
}
```

#### **Задание 4.1.5. Новый модуль спутника**

*Условие:*

Вам нужно создать новый модуль связи. Для того, чтобы интегрировать новый модуль в уже существующий спутник, необходимо разобраться как работает шина передачи данных спутника. В конструкторе все модули соединены по шине RS-485. Чтобы сообщения не терялись, есть один единственный мастер шины, который опрашивает все остальные устройства. Более подробно протокол описан в технической документации спутника. Ваша задача – прослушать шину и попытаться отделить сообщения для различных модулей.

*Используемое оборудование:*

Конструктор для сборки модели спутника, плата arduino mega, шилд-адаптер rs-485, ноутбук.

*Критерии оценки:*

- Вы можете увидеть все сообщения в шине в том виде, как они передаются, пересылая их в консоль компьютера – **3 балла**.
- Можете разделить сообщения начиная каждое новое сообщение с новой строки - **5 баллов**.
- Можете расшифровать тип сообщения – после каждого сообщения в шине программа подключенного модуля в консоли пишет тип и номер устройства к которому оно адресовано **10 баллов**.

- Можете выделить только сообщения адресованные УКВ передатчику №2 (в консоль выводятся только строки, которые вы передаете на УКВ передатчик №2) **10 баллов.**

*Комментарий:*

Для выполнения задания нужно понять суть работы протокола COBS в шире спутника. Подробное разъяснение команд мастера шины и ответов на них содержится в сопроводительном документе cobs.pdf

*Решение:*

```
void setup()
{
  Serial.begin(115200);
  Serial1.begin(115200);
}

void loop()
{
  if (Serial1.available() > 0) {
    Serial.write(Serial1.read());
  }
}
```

#### **Задание 4.1.1. Канал телеметрии**

*Условие:*

Для того, чтобы узнать о нормальной работе спутника, используется канал телеметрии. Это радиоканал, в который передается информация о состоянии бортовых систем. Вам необходимо составить отчет о состоянии бортовых систем, и отправить его на Землю. В телеметрии нужно указать Название вашего аппарата, номер отсылаемого сообщения, время в секундах прошедшее от запуска программы, скорость вращения аппарата. Данные нужно отсылать 1 раз в секунду

*Используемое оборудование:*

Конструктор для сборки модели спутника, ноутбук.

*Критерии оценки:*

- В телеметрии передается Название команды - **3 балла**
- Номер посылаемого пакета - **2 балла**
- Время в секундах прошедшее от запуска программы – **2 балла**
- Скорость вращения аппарата – **3 балла**
- Все данные собраны в строку, правильного формата, данные разделены символом « ; » **10 баллов**

*Комментарий:*

Для успешного выполнения задания, нужно передать строку, содержащую нужные параметры. Пример кода, передающего строку можно найти в примерах.

```
#include <stdio.h>
/* ** Lab 1: hello, world! */
void control(void) {
  int count = 0;
  puts("command_name");
  puts(count);
}
```



```

puts ("");
puts (TIME);
}

```

### Задание 4.1.1. Миссия

*Условие:*

Данные полученные от неизвестной планеты очень интересны, но большое количество шумов в радиоэфире не даёт связываться со спутником по имеющимся средствам связи. Вам нужно собрать аппарат, который сможет выполнить исследовательскую миссию по наблюдению за планетой и передать результаты наблюдений через созданный вами канал связи.

*Используемое оборудование:*

Конструктор для сборки модели спутника, модель планеты, платы arduino mega, Arduino Leonardo, шилд-адаптер rs-485, беспаячная макетная плата - 2 шт, модуль с ИК-светодиодом, ИК-приёмник TSOP4856, набор проводов для макетной платы, ноутбук 2 шт

*Критерии оценки:*

Спутник находится на орбите планеты, с него нужно уверенно получать телеметрию, 1 раз в секунду. Телеметрия передается через созданный вами оптический канал. В телеметрии нужно указать: Название вашего аппарата, номер отсылаемого сообщения, время в секундах прошедшее от запуска программы, скорость вращения аппарата, самое новое принятое сообщение от планеты, долготу подспутниковой точки. Данные должны быть разделены символом « ; ».

- Оценивается количество верно полученных пакетов, содержащих корректные данные в течении 25 секунд **1 балл** за каждый пакет (максимум **25 баллов**).
- На спутнике есть файл с фотографией, полученной камерой спутника. Вам нужно передать через созданный вами канал это изображение и показать его на принимающем компьютере. **25 баллов** пропорционально проценту искажений и потере. Максимальное время передачи 600 секунд.

*Комментарий:*

Суть этого задания – сборка всех предыдущих этапов в одно единое решение. Нужно стабилизировать вращение спутника, переслать со спутника данные на передающий контроллер, и принять их на принимающей стороне. Для приема был дан скрипт, который отображает принятые в порт данные в виде картинки. В качестве отсылаемого изображения использовался логотип олимпиады НТИ в формате bmp размером 100X100 точек.

Код приемной части аналогичен коду в задании 2, код передающей части аналогичен коду задания 2 с добавлением исходной информации из задания 4.

Код декодирующего скрипта в среде processing

```

import processing.serial.*;

PImage img;
Serial myPort;
int val;
int count = 0;

void setup() {
  size(100, 100);

```

```

    printArray(Serial.list());
    String portName = Serial.list()[3];
    myPort = new Serial(this, portName, 115200);
    img = createImage(100, 100, ALPHA);
}

void draw() {
    background(0);
    while ( myPort.available() > 0) { // If data is available,
        val = myPort.read(); // read it and store it in val
        // print(count); print(" "); println(val);
        img.pixels[count] = color(val);
        count++;
        if (count >= img.pixels.length) {
            count = 0;
        }
    }
    img.updatePixels();
    image(img, 0, 0);
}

```