

## §2 Второй отборочный этап

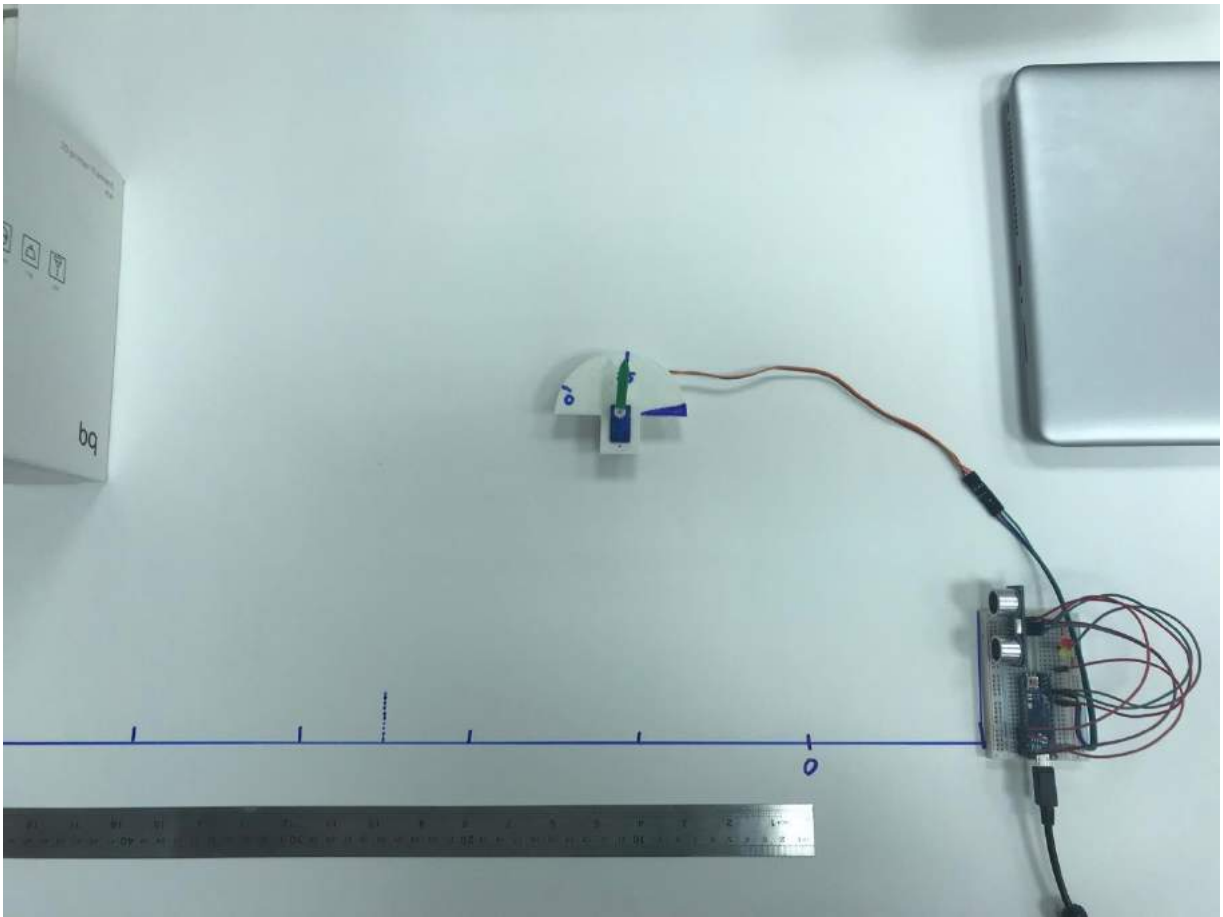
Второй отборочный этап проводится в командном формате в сети интернет, работы оцениваются вручную методистом в бланках оценки, весь процесс полностью фиксируется посредством видеозаписи тестов приемки, описанных в задании. Продолжительность второго отборочного этапа — 1 месяц и 1 неделя с учетом того, что задачи открываются участникам в онлайн-системе Stepik последовательно, первая задача была закрыта за 3 дня до окончания этапа, вторая задача была открыта на последние 10 дней. Содержание задач было открыто всем участникам, в то время как загрузка решений была доступна только для капитанов сформированных команд (по 3-4 человека). Задачи для 9-11 классов носят междисциплинарный характер и в более простой форме воссоздают инженерную задачу заключительного этапа. Решение каждой задачи дает определенное количество баллов. Баллы зачисляются в соответствии с объемом правильно прошедших тестов внутри задачи, для каждой задачи было две попытки, итоговая оценка считалась как сумма медиан баллов за попытки каждой задачи. В данном этапе можно получить суммарно от 0 до 39 баллов.

### 2.1 Задача №1 (18 баллов макс.)

Перед вами задача для среды разработки Arduino IDE. Обратите внимание что задачи программно-аппаратные, и будут проверяться на реальном стендовом оборудовании собранным в соответствии со схемой и с использованием указанных компонентов. Нельзя использовать библиотеки сверх тех, что указаны в задаче.

Задание:

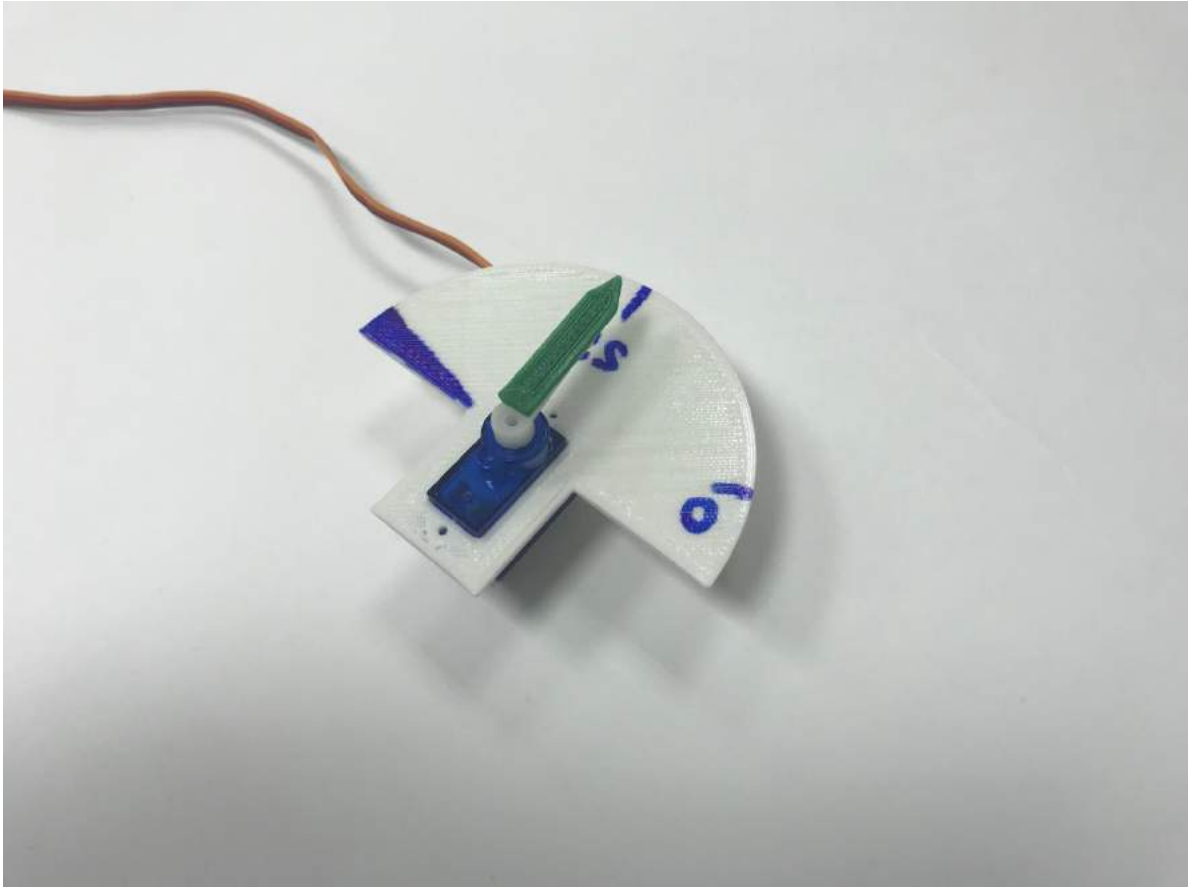
Дан стенд, измеряющий расстояние на основе ультразвукового датчика и выводящий это расстояние на стрелочный индикатор, сделанный на основе стандартного микро сервопривода. Диапазон работы сервопривода - от 20 до 160 градусов. Измеряемое расстояние составляет 50 сантиметров. Нулевая отметка установки находится на расстоянии 10 см от датчика (см. Фото 1). Кроме этого, на стенд выведены два светодиода - желтый и красный (назначение описано ниже).



*Фото 1 - Вид стенда сверху, по горизонтали - рабочая шкала для измерений*

Перед датчиком рукой перемещается преграда. Необходимо написать код программы, реализующей следующий алгоритм и удовлетворяющей требованиям:

1. Если устройство видит преграду на расстоянии от 0 до 50 см, она выводит это расстояние на стрелочный индикатор, в соответствии с калибровкой: 0 см - 160 градусов, 50 см - 20 градусов (см. Фото 2).



*Фото 2 - Вид стрелочного индикатора на базе сервопривода*

2. Если расстояние между преградой и датчиком увеличивается, загорается красный светодиод (см. Фото 3 и Схему 1). Если расстояние уменьшается - горит желтый светодиод. Если в начальном положении состояние “не определено”, то считаем, что расстояние увеличивается. Если преграда не двигается, выводим последнее определенное состояние.
3. Если преграда отсутствует или выходит за пределы измерений, загораются оба светодиода и индикатор перемещается в положение «0». Если система не меняет свое состояние "вне границ", то оба светодиода продолжают светиться.

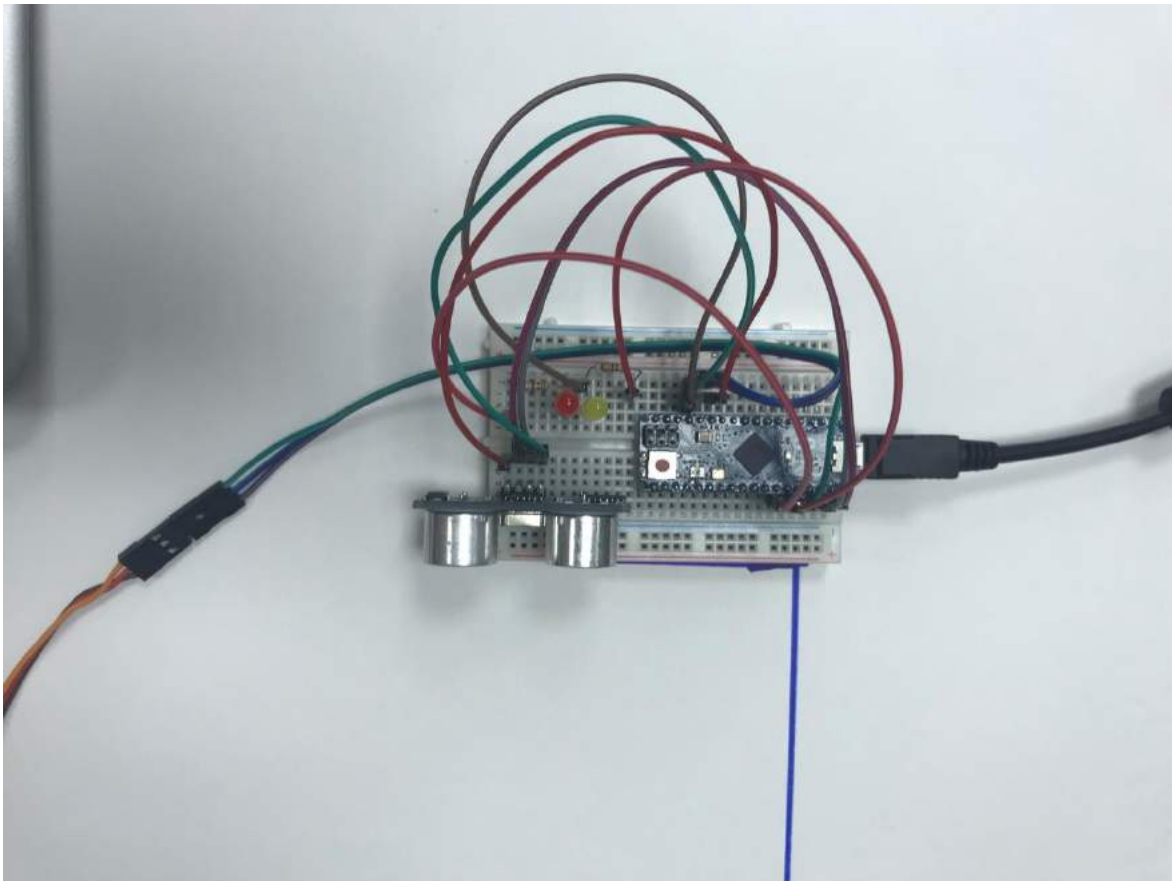


Фото 3 - Вид на подключенные светодиоды сверху

4. Программа работает стабильно (перебор разных комбинаций действий с пластиной, описанных в пп. 1-3 в течении 30 секунд, всегда воспроизводится).
5. Задержка “реакции” индикатора на перемещение преграды должна быть не более 0.5 сек.
6. Значения, выводимые на индикатор должны усредняться и округляться - стрелка не должна дребезжать если пластина не двигается. Предполагаем, что шум датчика достигает  $\pm 1$  см.

Используемое оборудование в рамках стенда:

- Arduino micro R3 (на базе микропроцессора процессора Atmega32U4)
- Сонар HC-SR04
- Микро-сервопривод TowerPro SG90

Схема подключений устройств дана ниже, на Схеме 1 (Dn - номер цифрового входа-выхода Ардуино, An - номер аналогового входа-выхода Ардуино).

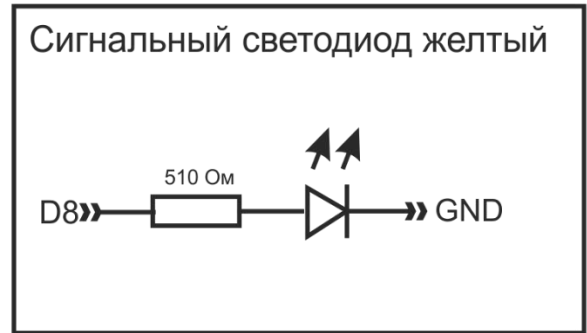
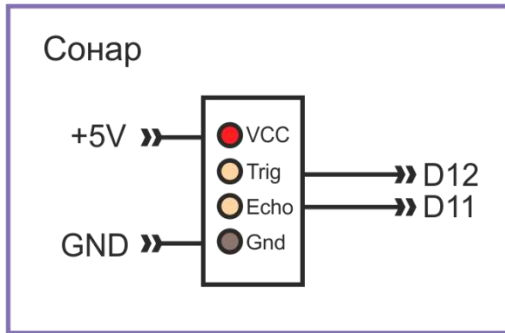
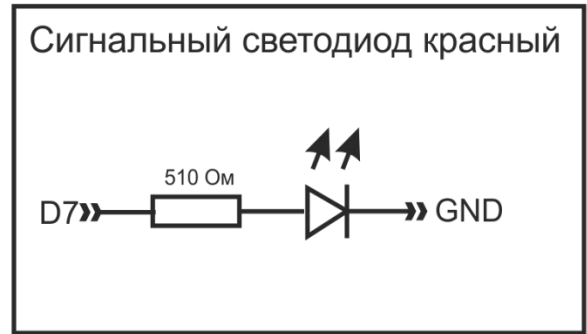
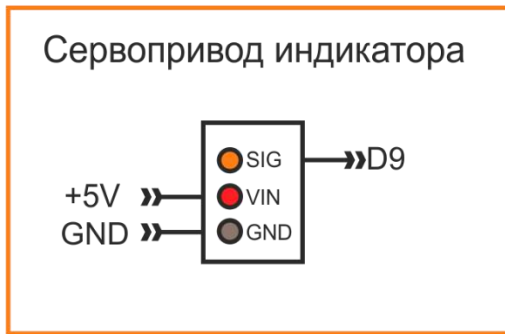


Схема 1 – схема подключения оборудования

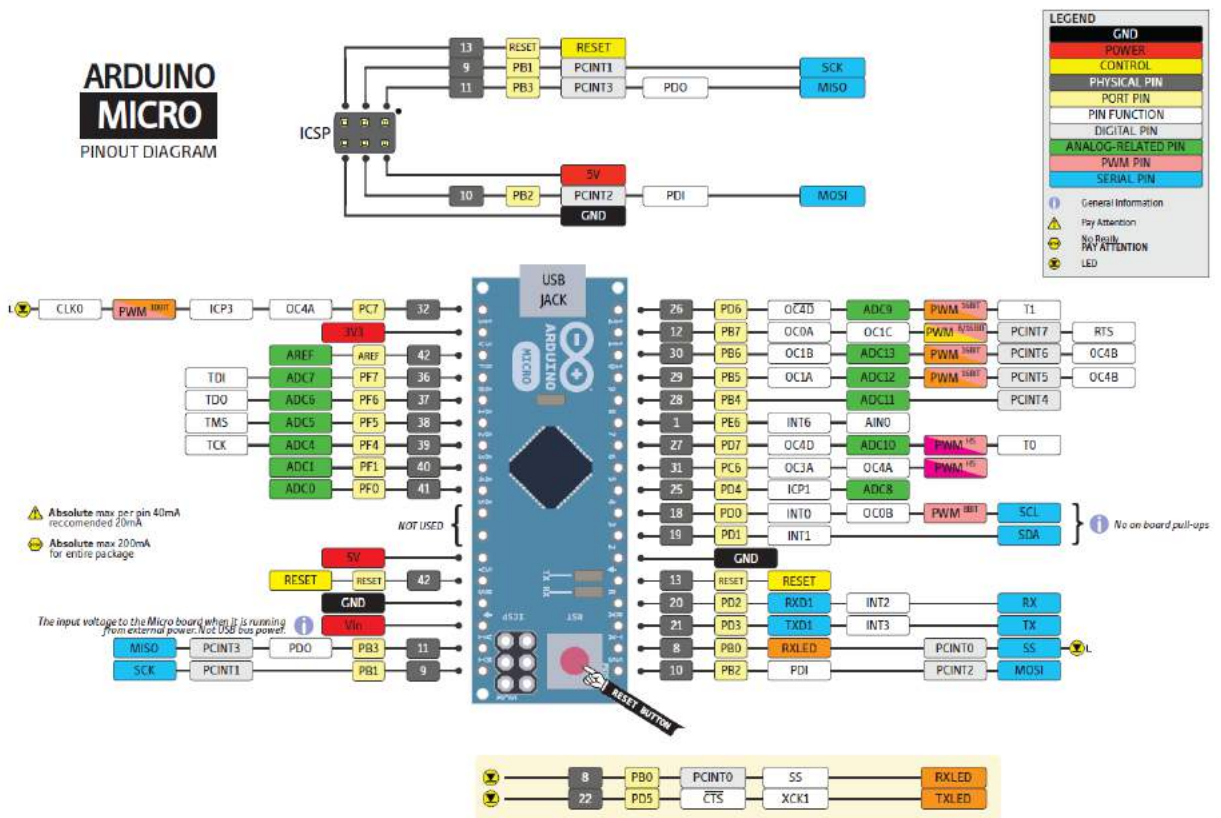


Схема 2 – принципиальная схема Arduino Micro R3

Пример работы стенда доступен по ссылке: <https://www.youtube.com/watch?v=I9sbxGC3W1o>  
 Формат ответа:

Задача должна быть представлена в виде одного файла с программой. Файл должен называться по фамилии участника, загружающего файл на рецензию (капитана команды), быть написан латинскими буквами и иметь в названии номер попытки. Например, "Ivanov-1.ino". Кроме того, в заголовке программы необходимо указать ФИО участника (оно должно совпадать с именем файла), версию (номер попытки) и название команды. Опционально можно указать ФИО автора и других членов команды). Для получения обратной связи тут же в заголовке необходимо указать адрес электронной почты. Всего за задание могут быть начислены 18 баллов. Могут быть начислены 2 штрафных балла за несоответствие требованиям к оформлению решений.

Возможно использование библиотек (они будут установлены на проверочном стенде):

1. NewPing (v1.8) (Доступна по ссылке: [https://bitbucket.org/teckel12/arduino-new-ping/downloads/NewPing\\_v1.8.zip](https://bitbucket.org/teckel12/arduino-new-ping/downloads/NewPing_v1.8.zip))
2. Servo (Встроенная в рамках среды разработки Arduino IDE)

Методика проверки задачи и критерии:

- Загрузка на стенд вашей программы через среду Arduino IDE 1.8.5.
- Проверка работоспособности согласно Бланку проверки.
- Каждый пункт бланка проверки независим и приносит 1 балл.
- Происходит видеофиксация попытки.
- Заполненный бланк (см. Таблица 1) и видео загружаются как рецензия к решению в рамках текущей попытки.

## Бланк проверки

Команда:	
Участник:	
Дата:	
Попытка:	
Имя файла:	

Если программа выдает ошибки при компиляции или загрузке и это не связано с отсутствием указанных в задании библиотек, расположением и другими не связанными с кодом причинами, то попытка получает **0 баллов!**

Программа компилируется и загружается	
---------------------------------------	--

Перед началом проверки контроллер перезагружается

1	Препятствие ставится на середину пути (25 см)	
	>	
2	Препятствие плавно двигается в сторону отметки "0"	
	> Стрелка индикатора движется в сторону отметки "0" до её достижения	
	> Стрелка индикатора останавливается в отметке "0"	
	> При движении горит желтый светодиод	
	> При движении отсутствует "мигание" светодиодов	
3	Препятствие плавно двигается в сторону отметки "50"	
	> Стрелка индикатора движется в сторону отметки "50" до её	
	> Стрелка индикатора останавливается в отметке "50"	
	> При движении горит красный светодиод	
	> При движении отсутствует "мигание" светодиодов	
4	Препятствие выезжает за границу 50	
	> Стрелка вернулась на отметку "0"	
	> Горят два светодиода	
5	Препятствие убирается	
	> Состояние стрелки и светодиодов без изменений (статично во время убирания)	
6	Препятствие появляется на расстоянии меньше отметки "0"	
	> Состояние стрелки и светодиодов без изменений (статично во время убирания)	
7	Препятствие появляется на расстоянии "25"	
	> Стрелка индикатора указала на правильную отметку "25"	
8	Препятствие ставится в произвольное место от 0 до 50 и стоит там в течении 5 секунд.	
	> Стрелка переходит в соответствующий сектор шкалы	
	> В течении 5 секунд отсутствует дрожание стрелки	
9	Препятствие двигается взад-вперед по шкале 2 раза с выходом за границы измерений	
	> Светодиоды работают корректно (при увеличении красный, при уменьшении желтый, за границами оба)	
	> Стрелка индикатора работает корректно (показывает соответствующие значения, верно обрабатываются границы)	

Таблица 1 - Бланк проверки задания

**Решение:**

```
#include <Servo.h>
```

```

#define PIN_TRIG 12

#define PIN_ECHO 11

#define MIN_DISTANCE 10

#define MAX_DISTANCE 50 + MIN_DISTANCE

#define RED_PIN 7

#define YELLOW_PIN 8

#define SERVO_PIN 9

#define MIN_ANGLE 20

#define MAX_ANGLE 160

Servo myservo;

long Value, OldValue=0;

long v1,v2,v3,v4,v5=0;

int pos;

long Mediana(const long a, const long b, const long c) // среднее из трех значений
{
    long v;

    if (a>=b) {
        if (b>=c)
            {v=b;}
        else
            if (a>=c)
                {v=c;}
            else {v=a;};
    }
    else {
        if (a>=c)
            {v=a;}
    }
}

```



```

else {
    if (b>=c)
        {v=c;}
    else {v=b;};
};
return v;
}

```

```

long GetByPulse() {
    digitalWrite(PIN_TRIG, HIGH);

    // Выставив высокий уровень сигнала, ждем около 10 микросекунд. В этот момент датчик
    // будет посылать сигналы с частотой 40 КГц.

    delayMicroseconds(10);

    digitalWrite(PIN_TRIG, LOW);

    long duration = pulseIn(PIN_ECHO, HIGH);

    long cm = (duration / 2) / 29.1;

    return cm;
}

```

```

void GetValue() {
    long cm = GetByPulse();

    delay(100);

    if ( (cm > 0) and (abs(OldValue-cm)>1) ) {
        Value=cm;
    }

    else Value=OldValue;
}

```

```

void ShowValue() {
    if ((OldValue-Value)==0) {
        //digitalWrite(YELLOW_PIN, HIGH);
    } else {
        if ((Value<MIN_DISTANCE)or(Value>MAX_DISTANCE)) {
            digitalWrite(RED_PIN, HIGH);
            digitalWrite(YELLOW_PIN, HIGH);
            pos=MAX_ANGLE;
        }
        else {
            if (Value > OldValue) {
                digitalWrite(RED_PIN, HIGH);
                digitalWrite(YELLOW_PIN, LOW);
            } else if (Value < OldValue) {
                digitalWrite(RED_PIN, LOW);
                digitalWrite(YELLOW_PIN, HIGH);
            };
            pos=MAX_ANGLE - (Value-MIN_DISTANCE)*(MAX_ANGLE-
MIN_ANGLE)/(MAX_DISTANCE-MIN_DISTANCE);
        };
        myservo.write(pos);

        Serial.print (Value);
        Serial.print ("cm Угол=");
        Serial.print ln(pos);

        OldValue = Value;
    }
}

```

```
}
```

```
void Filtration() {  
    v1=OldValue; v2=v3; v3=v4; v4=v5; v5=Value;  
  
    long s1,s2,s3;  
  
    s1=Mediana(v1,v2,v3);  
  
    s2=Mediana(v2,v3,v4);  
  
    s3=Mediana(v3,v4,v5);  
  
    Value=Mediana(s1,s2,s3);  
}
```

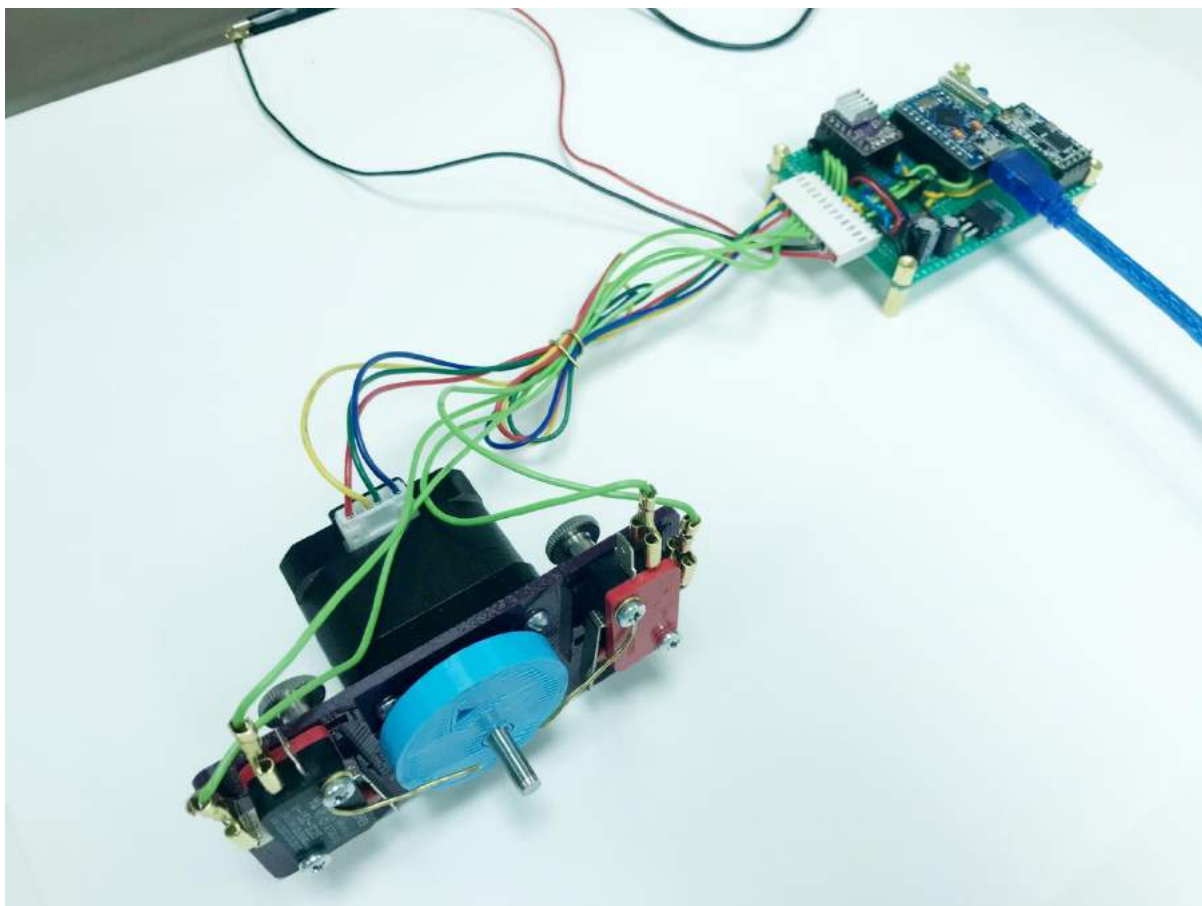
```
void setup() {  
    myservo.attach(SERVO_PIN);  
  
    Serial.begin(9600);  
  
    pinMode(PIN_TRIG, OUTPUT);  
  
    pinMode(PIN_ECHO, INPUT);  
  
    pinMode(RED_PIN, OUTPUT);  
  
    pinMode(YELLOW_PIN, OUTPUT);  
  
    OldValue=v1=v2=v3=v4=v5=MIN_DISTANCE;  
}
```

```
void loop() {  
    GetValue();  
  
    Filtration();  
  
    ShowValue();  
}
```

## 2.2 Задача №2 (21 балл макс.)

Перед вами задача для среды разработки Arduino IDE. Обратите внимание что задачи программно-аппаратные, и будут проверяться на реальном стендовом оборудовании собранным в соответствии со схемой и с использованием указанных компонентов. Нельзя использовать библиотеки сверх тех, что указаны в задаче.

Задание: Дан испытательный стенд, представляющий собой систему из шагового двигателя с ассиметричным маховиком и системой приема данных через радиоэфир (см. Фото 4).



*Фото 4 – Испытательный стенд, взгляд сверху*

Необходимо написать программу, которая обеспечивает работу стенда следующим образом:

1. Постоянно, без остановок и пауз осуществляется колебательное движение маховика от левого концевого выключателя (№1) до правого концевого выключателя (№2). В крайнем правом положении мотор должен задерживаться на одну секунду. В крайнем левом положении мотор не задерживается - сразу начинается обратное вращение. При срабатывании правого концевика загорается синий светодиод. При срабатывании левого концевика синий светодиод гаснет и загорается желтый светодиод. Светодиоды продолжают гореть в установленном режиме пока не работает противоположный концевик.

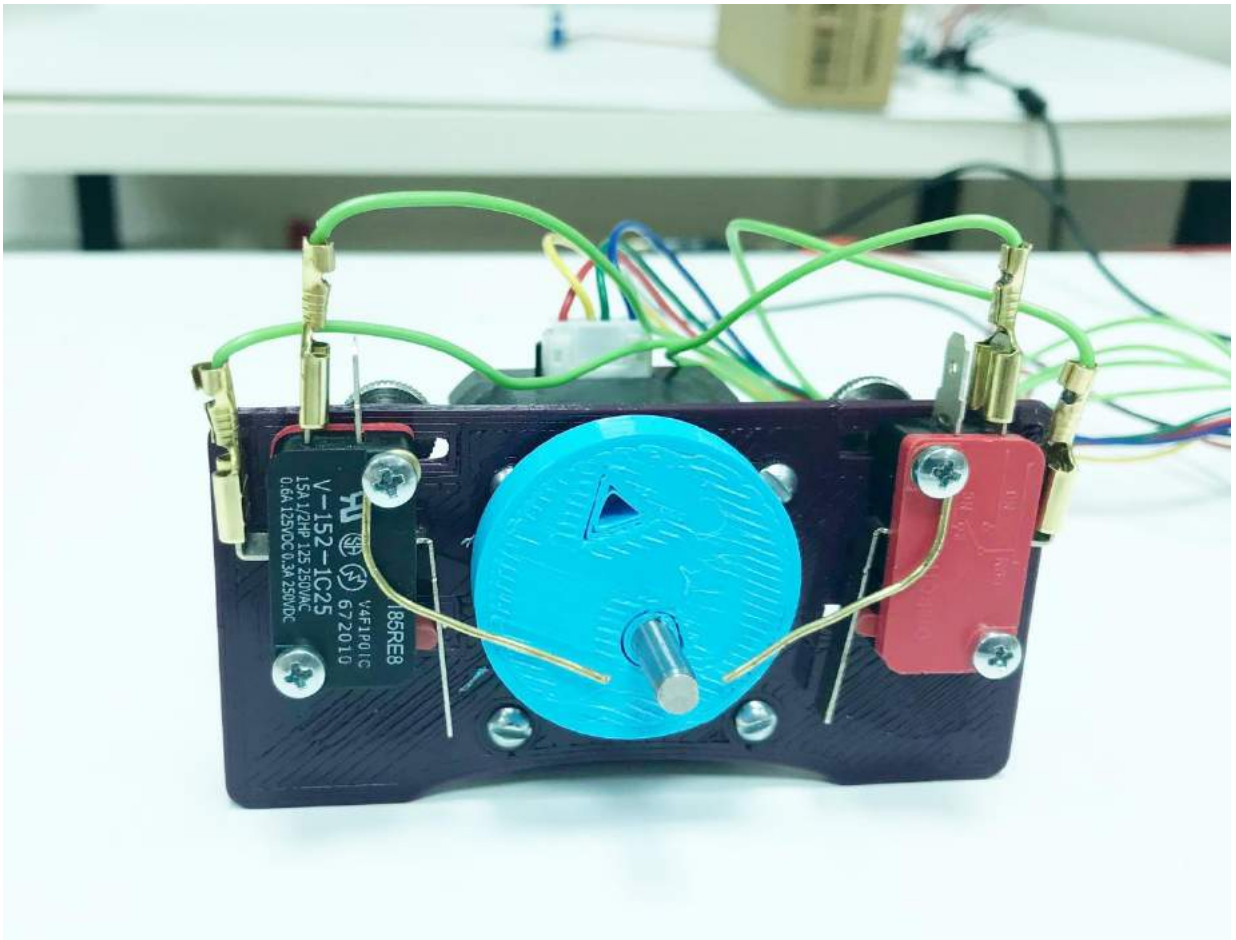


Фото 5 - Правый концевик на картинке - с красной крышкой. Проволоки - отметки ниже которых не должна уходить стрелка.

2. Скорость работы двигателя должна быть не ниже 500 шагов в секунду. Способ контроля: 10 циклов работы маховика должно происходить не более чем за 15 секунд.
3. Осуществляется прием без остановок и пауз и верификация поступающей на порт (Serial1) через радиомодуль информации.
4. К каждой корректной строке принятой информации добавляется порядковый номер начиная с "1" и пробел. Сформированная строка отправляется в последовательный порт (Serial).
5. Каждая некорректная строка засчитывается за ошибку, нумеруется и выдается в порт (Serial) сообщением "{номер ошибки} Error / {полученный текст}"
6. Все передачи идут по строкам, и заканчиваются концом строки (\n)

Корректные фразы:

- "The quick brown fox jumps over the lazy dog"
- "Beyond font testing, these tools can also be useful"

Дополнительные условия:

- Фразы приходят подряд построчно. Количество ошибочных фраз перед каждой проверкой изменяется и в ходе проверки сравнивается полученное вами число.
- Скорость работы радиомодуля на порту Serial1 - 1200bps
- Скорость работы порта Serial - 9600bps

Возможно использование библиотек (они будут установлены на проверочном стенде):  
<https://github.com/sadr0b0t/arduino-timer-api>

Используемое оборудование в рамках стенда (подключение см. на Схемах 3-6 и на Фото 6):

- Arduino Pro micro (Atmega 32U4), аналогично Arduino Leonardo
- Двигатель шаговый Nema 17
- Драйвер шагового двигателя DRV8825 (Для вращения вправо DIR должен быть LOW, Полную, документацию необходимо найти самостоятельно)
- Радиомодуль HC12 (Скорость передачи настроена, работает с портом Serial1. Полную документацию необходимо найти самостоятельно, краткая доступна по ссылке: <https://yadi.sk/i/mvroui9h3QFs9d>)
- Концевые выключатели, нормально разомкнутые (Левый - №1, Правый №2, смотря со стороны маховика)
- Светодиод Синий - (Светодиод №1)
- Светодиод Желтый - (Светодиод №2)

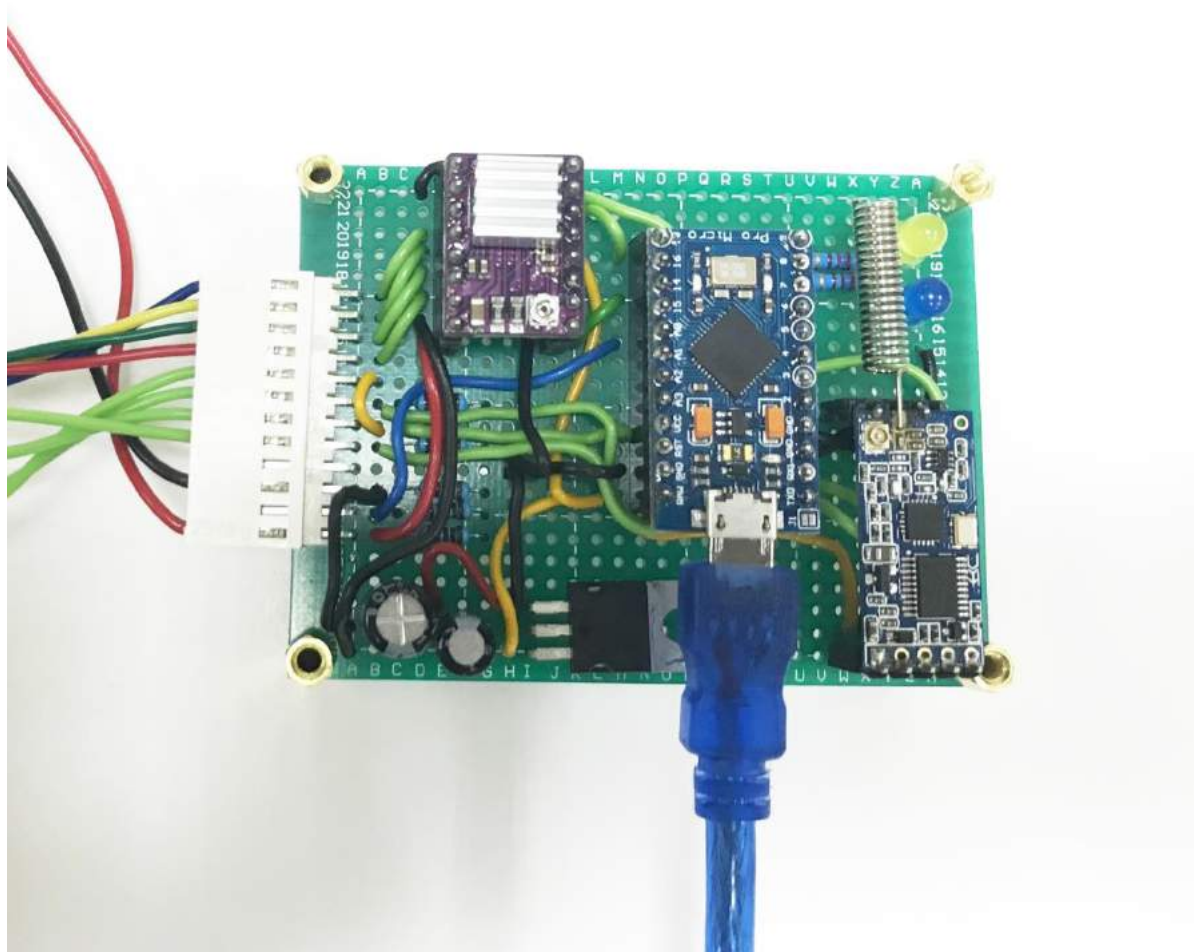


Фото 6 – Подключение к плате, вид сверху

## Модуль управления двигателем DRV8825

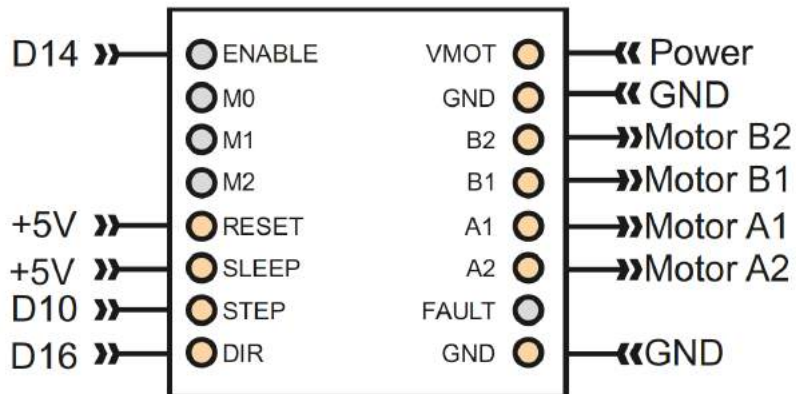


Схема 3 – Модуль управления шаговым двигателем, подключение

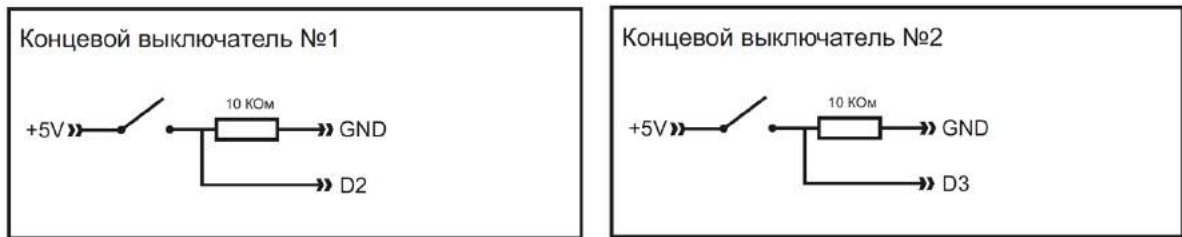


Схема 4 - Схема подключения, концевые выключатели

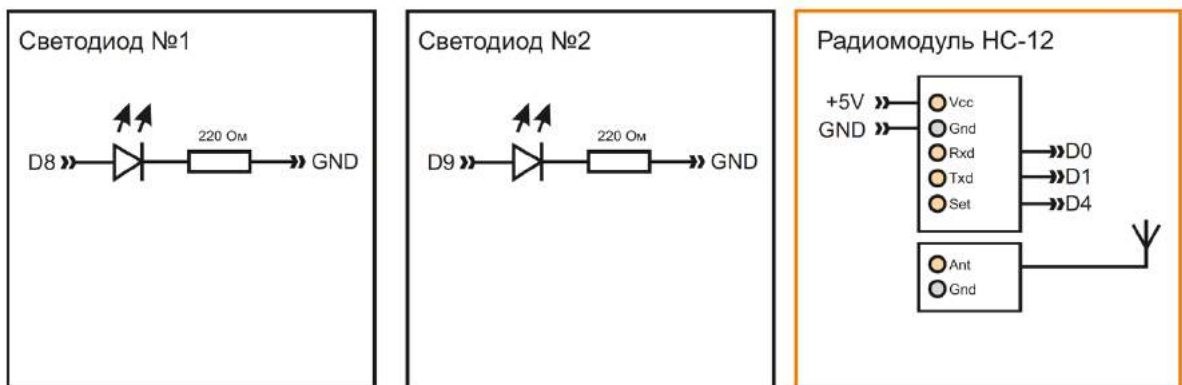


Схема 5 – Подключение к периферии



# PROMICRO

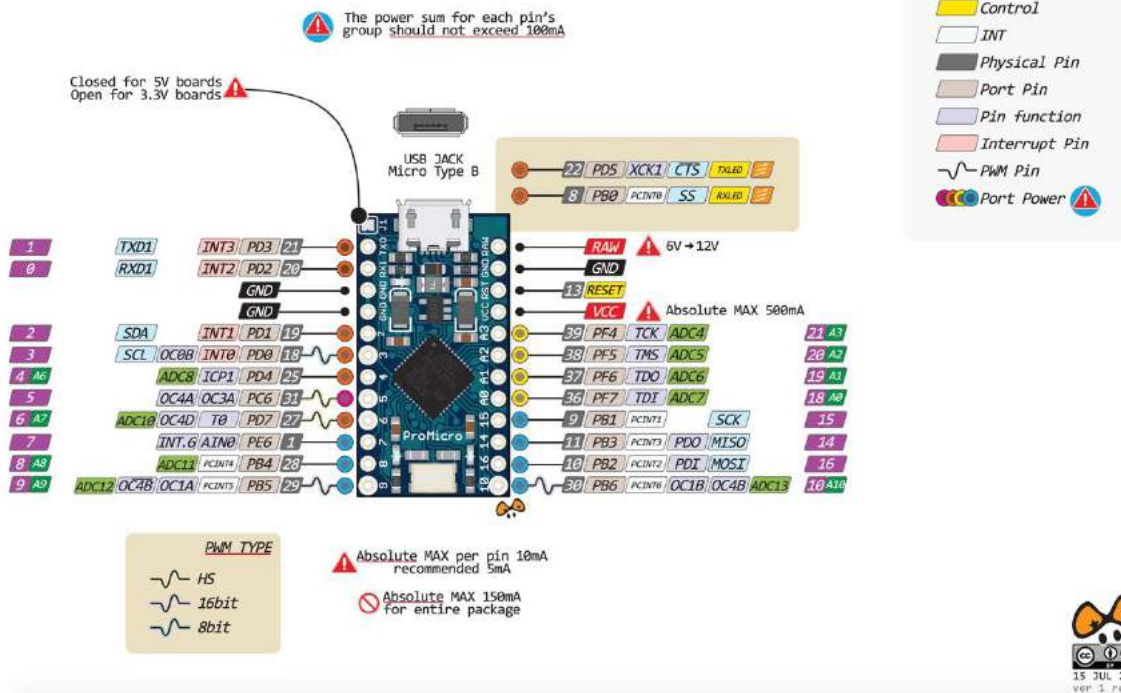


Схема 6 – Выводы Arduino ProMicro

Пример работы стенда доступен по ссылке: <https://www.youtube.com/watch?v=u7kX6OQ8AA4>

Формат ответа: Задача должна быть представлена в виде одного файла с программой. Файл должен называться по фамилии участника, загружающего файл на рецензию (капитана команды), латинскими буквами и иметь в названии номер попытки. Например, "Ivanov-1.ino". Кроме того, в заголовке программы необходимо указать ФИО участника (фамилия должна совпадать с именем файла), версию (номер попытки) и название команды. Опционально можно указать ФИО автора и других членов команды). Для получения обратной связи тут же в заголовке необходимо указать адрес электронной почты. Всего за задание может быть начислен 21 баллов. Могут быть начислены 2 штрафных балла за несоответствие требованиям к оформлению решений.

Методика проверки задачи и критерии: Загрузка на стенд вашей программы через среду Arduino IDE 1.8.5. Проверка работоспособности согласно Бланку проверки (см. Таблица 2). Каждый пункт бланка проверки независим и приносит 1 балл. Происходит видеofиксация попытки.



### Бланк проверки задачи №3

Команда:	
Участник:	
Дата:	
Попытка:	
Имя файла:	

Если программа выдает ошибки при компиляции или загрузке и это не связано с отсутствием указанных в задании библиотек, расположением и другими не связанными с кодом причинами, то попытка получает **0 баллов!**

Программа компилируется и загружается	
---------------------------------------	--

Маховик ставится отметкой вверх (стрелка на диске), питание двигателя отключено, плата Ардуино подключена к компьютеру.	
1	После прошивки светодиоды не загораются (небыло нажатий концевиков)
2	Подаем питание. Мотор начал вращаться
3	Мотор начал вращаться в правую сторону
4	Маховик нажал на правый концевик и остановился
5	Маховик остановился в обозначенных рамках (треугольник выше отметки)
6	Маховик остановился на одну секунду
7	Маховик начал движение в обратную сторону
8	Маховик нажал на левый концевик и не остановился (сразу поменял направление)
9	Смена движения маховика произошла в обозначенных рамках (треугольник выше отметки)
10	Маховик начал движение в обратную сторону
11	Скорость работы двигателя должна быть не ниже 500 шагов в секунду. Способ контроля: 10 циклов работы маховика должно происходить не более чем за 15 секунд. При невыполнении этого пункта начисляется 5 штрафных баллов
12	Корректно работают светодиоды
13	Цикл работы программы с маховиком работала более <b>10 секунд</b> от старта без ошибок (проскакиваний концевиков, остановок, изменения скорости маховика)
14	Тестовые фразы принимаются и выводятся в порт (проверка программой монитор порта)
15	Корректные фразы нумеруются
16	Ошибочные фразы обрабатываются корректно
17	Ошибочные фразы корректно маркируются
18	В конце работы программы число ошибок правильное
19	В конце работы программы число корректных фраз правильное
20	При передаче данных отсутствовали задержки (регулярный одинаковый интервал между посылками)
21	Цикл работы программы с маховиком работала <b>все время тестирования</b> (проскакиваний концевиков, остановок, изменения скорости маховика)

Таблица 2 – Бланк проверки задачи №2

Решение:

```
#include"timer-api.h"
```

```
#define STEP_PIN 10

#define DIR_PIN 16

#define EN_PIN 14

#define leftButton 1

#define rightButton 0

#define blueLed 8

#define yellowLed 9

int stateRotate = 0;

const int dl = 1000;

unsigned long tm = 0;

unsigned long goodStr = 1;

unsigned long badStr = 1;

String str = "";

const String normStr[2] = {"The quick brown fox jumps over the lazy dog",
                          "Beyond font testing, these tools can also be useful"};

void setup() {

  Serial.begin(9600);

  Serial1.begin(1200);

  pinMode(4, OUTPUT);
```

```

digitalWrite(4, HIGH); //set

pinMode(EN_PIN, OUTPUT);
digitalWrite(EN_PIN, LOW);

pinMode(STEP_PIN, OUTPUT);
digitalWrite(STEP_PIN, LOW);

pinMode(DIR_PIN, OUTPUT);
digitalWrite(DIR_PIN, LOW);

attachInterrupt(leftButton, clickLeftButton, RISING);
attachInterrupt(rightButton, clickRightButton, RISING);

pinMode(blueLed, OUTPUT);
pinMode(yellowLed, OUTPUT);
digitalWrite(blueLed, LOW);
digitalWrite(yellowLed, LOW);

timer_init_ISR_500Hz(TIMER_DEFAULT); //500Hz
}

void loop() {

    if (Serial1.available()) {
        char a = (char)Serial1.read();
    }
}

```

```

if (a == '\n') {
  if (str == normStr[0] || str == normStr[1]) {
    Serial.print (goodStr);
    Serial.print (" ");
    goodStr++;
  } else {
    Serial.print (badStr);
    Serial.print (" Error /");
    badStr++;
  }
  Serial.print ln(str);
  str = "";
} else
  str += a;
}

```

```

if (stateRotate == 2) {
  tm = millis() + dl;
  digitalWrite(EN_PIN, HIGH);
  stateRotate = 1;
}

```

```

if (tm <= millis())
  digitalWrite(EN_PIN, LOW);

```

```

digitalWrite(DIR_PIN, stateRotate);

}

void timer_handle_interrupts(int timer) {

    digitalWrite(STEP_PIN, HIGH);

    delayMicroseconds(1);

    digitalWrite(STEP_PIN, LOW);

}

void clickLeftButton(){

    stateRotate = 0;

    digitalWrite(blueLed, LOW);

    digitalWrite(yellowLed, HIGH);

}

void clickRightButton(){

    stateRotate = 2;

    digitalWrite(blueLed, HIGH);

    digitalWrite(yellowLed, LOW);

}

```

### **2.3 Критерии определения победителей и призеров\***

Итоговый рейтинг формируется из первых 38 человек в составах команд, набравших максимальную оценку исходя из суммарного балла команды, составленного из сумм медиан баллов за попытки каждой задачи. Призерам второго отборочного этапа было необходимо набрать 29 баллов из 39 возможных. Победители второго отборочного этапа - 3 команды в рейтинге, набравших максимальный балл.

\* Также в финал приглашаются победители и призеры Олимпиады НТИ-2017 (4 человека) и открытых публичных конкурсов, список которых утвержден Оргкомитетом (профильные разделы конкурсов: хакатон “GoToHack” инженерно-конструкторская школа “Лифт в будущее”, форум “Будущие интеллектуальные лидеры России, национальный этап Worldskills) (1 человек). Оргкомитет связывается с лицами, имеющими право участия в финале индивидуально и запрашивает подтверждение их участия в финале.