

Олимпиада «Высшая проба» проводится при поддержке Сбера, приветствуем участников соревнования!



Сбер – лидер по созданию инноваций, которые эффективно интегрируются в жизнь людей.

Поздравляем – ты являешь участником заключительного этапа олимпиады по направлению «Инженерные науки».

Вперед к наукоемким технологиям! Желаем тебе блистательной победы!

Время выполнения – 200 минут.

Пишите разборчиво. Ответ пишите на странице с соответствующим номером вопроса. Если используете дополнительный лист, обязательно напишите об этом на основном листе ответа. Если не знаете ответа, ставьте прочерк. Черновики не оцениваются.

Нахождение каждого ответа должно сопровождаться подробной аргументацией. Участником должны быть приведены необходимые для объяснения логики решения рисунки, формулы, аналитические обоснования.

Приступая к выполнению заданий, вы подтверждаете, что профиль и класс заданий соответствуют сведениям, указанным вами при регистрации.

(Максимальное количество баллов – 100)

Задача 1 (Максимум – 25 баллов).

Четыре ретранслятора связи №1, №2, №3 и №4 образуют ориентированную сеть. Наличие канала задано логическими высказываниями: если высказывание истинно - ребро существует, ложно - ребра нет. Дано: $A=1$, $B=0$. Операции: $*$ — И, $\oplus$ — XOR, $\rightarrow$ — импликация, $\neg$ — НЕ.

Матрица (логические высказывания):

	1	2	3	4
1	$(A*B)\oplus(\neg A)$	$(A\rightarrow B)\oplus A$	$\neg(A\oplus B)$	$(\neg B)\rightarrow A$
2	$(A\rightarrow B)*\neg B$	$(A*B)\oplus B$	$(A\oplus B)\rightarrow A$	$\neg(A*B)$
3	$(A*B)\rightarrow(A\oplus B)$	$(\neg B)\oplus(A\rightarrow A)$	$(A*B)\oplus(A\rightarrow B)$	$(A\oplus B)*\neg A$
4	$\neg B$	$(A\rightarrow B)\oplus(\neg B)$	$(A\oplus B)\oplus(A*B)$	$(A*B)\rightarrow B$

Если ребро существует, то его вес (в часах) записан в указанной системе счисления. Для отсутствующих рёбер вес равен 0.

Матрица весов:

	1	2	3	4
1	0	111001_2	0	20_8
2	0	0	57_8	10111_2
3	$2D_{16}$	0	0	0
4	11010_2	$3A_{16}$	52_8	0

Ответьте на вопросы:

- 1) По логическим высказываниям определите количество ориентированных рёбер в графе (без петель).
- 2) Переведите все ненулевые веса в десятичную систему и найдите максимальный вес ребра.
- 3) Найдите общий вес графа (сумма весов всех существующих рёбер в десятичной системе).
- 4) Определите минимальный вес пути из узла 1 в узел 4.
- 5) Определите максимальный вес пути из узла 2 в узел 4 по простым путям (без повторения вершин).

В ответы вносите целые числа, полученные при решении.

Задача 2 (Максимум – 25 баллов).

При разработке блока питания портативного медицинского устройства используется схема, представленная на рисунке 1. Номиналы сопротивлений $R1 = 5 \text{ кОм}$, $R2 = 3 \text{ кОм}$, $R3 = 2 \text{ кОм}$, $R4 = 4,5 \text{ кОм}$, $R5 = 2 \text{ кОм}$, $R6 = 3 \text{ кОм}$, $R7 = 6 \text{ кОм}$. Номиналы емкостей конденсаторов $C1 = 0,1 \text{ мкФ}$, $C2 = 0,02 \text{ мкФ}$. Индуктивность катушки $L1 = 0,4 \text{ мГн}$ и внутреннее сопротивление $RL1 = 2 \text{ Ом}$. Диод $D1$ считать идеальным. Напряжение источника ЭДС $U = 36 \text{ В}$. Ключ Key замкнут в первый момент времени ($t = 0 \text{ с}$). В рамках проверки стабильности питания необходимо определить значения электрических параметров схемы для установившегося режима (после завершения переходных процессов).

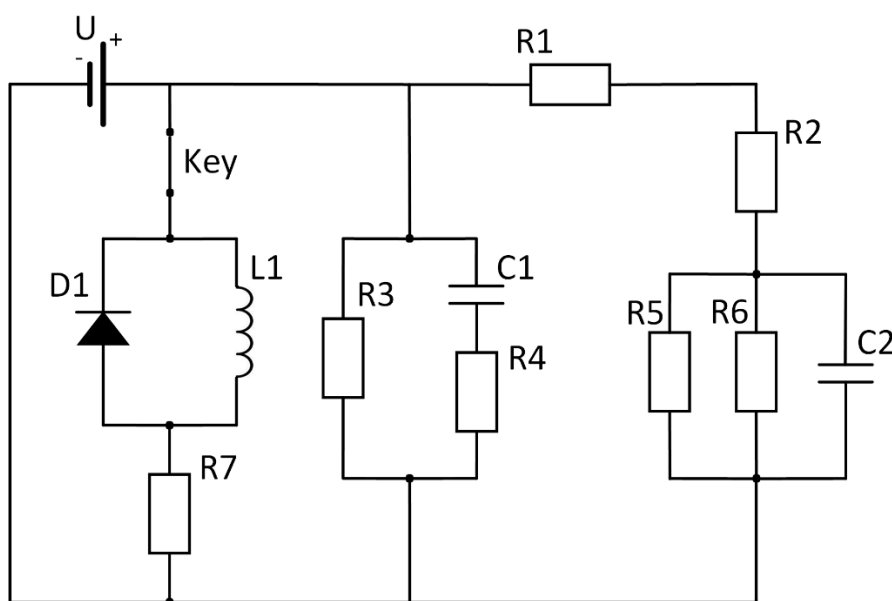


Рисунок 1. Электрическая схема.

Вопросы:

1. Определите значение тока через источник U для случая замкнутого ключа.
2. Определите ток через сопротивление $R5$ для случая замкнутого ключа.
3. Определите мощность ($P3$) выделяемую на сопротивление $R3$ для случая замкнутого ключа.
4. Определите заряд ($Q1$) на конденсаторе $C1$ для случая замкнутого ключа.
5. Определите энергию ($WL1$) катушки индуктивности $L1$ для случая замкнутого ключа.
6. Опишите какую функцию выполняет диод $D1$ в случае размыкания ключа в момент времени после завершения переходных процессов и имеет ли значение каким образом подключен диод

Задача 3 (Максимум – 25 баллов).

Для передачи данных о температуре океана на буйковых станциях используется инверсное помехоустойчивое кодирование. Температура измеряется с точностью 0.5°C (16 уровней от 0°C до 7.5°C) и представляется 4-битным словом a (от 0000 до 1111 соответственно).

При передаче по гидроакустическому каналу каждый бит может быть искажён помехой с вероятностью $q = 0.04$. Помехи в разных битах независимы.

Для повышения надёжности применяется кодер $K(a)$, который формирует 8-битное кодовое слово по правилу.

$$K(a) = a \& \begin{cases} a, & \text{если } \omega(a) \div 2 \\ \bar{a}, & \text{иначе} \end{cases}$$

Декодер определяет наличие помех в полученном значении b по принципу чётного веса. Если $\omega(b)$ нечётный, то считают, что произошла одна ошибка, на позицию которой укажет побитовая сумма половин полученного вектора. Допустим полученное кодовое слово b состоит из двух половин b_1 и b_2 . Тогда, анализируется 4-битная строка

$$D(b_1 \& b_2) = b_1 \text{ XOR } \begin{cases} b_2, & \text{если } \omega(b_1) \div 2 \\ \bar{b}_2, & \text{иначе} \end{cases}$$

- Если $\omega(D(b_1 \& b_2)) = 0$, то помех не было, и декодированным словом назовут b_1 .
- Если $\omega(D(b_1 \& b_2)) = 1$, то помеха пришлась на вторую половину и декодированным словом назовут b_1 .
- Если $\omega(D(b_1 \& b_2)) = 3$, то помеха пришлась на первую половину в позиции, где оказался 0 у $D(b_1 \& b_2)$. В b_1 меняют этот бит на противоположный и называют результат декодированным словом.
- Если $\omega(D(b_1 \& b_2))$, то было несколько помех, данная конструкция кода не восстанавливает такие случаи.

Задания:

- 1) Найдите вероятность того, что в переданном слове произойдёт ровно одна ошибка. Ответ округлите до трёх знаков после запятой.
- 2) Как выглядит $K(a)$ для температуры $6,5^{\circ}\text{C}$?
- 3) Декодируйте полученные данные от буя: 11100001 11001110 10100100
В ответе укажите последовательность принятых показателей температуры с точностью до полградуса (без указания единиц измерения), перечисленные в полученном порядке через запятую.
- 4) Получено кодовое слово 11001010. Сколько возможных комбинаций помех могло произойти в канале, если считать, что на одно кодовое слово приходится не более двух?

Комментарии:

- Назовем весом ω кодового слова a количество единиц в нём: $\omega(1001) = 2$.
- Назовём инверсией $\bar{a}$ кодового слова a битовую строку, в которой каждый бит кодового слова a заменён на противоположное значение. Например, $\overline{1001} = 0110$.
- Конкатенация кодовых слов a и b – операция склеивания: $a \& b$

- Побитовая сумма двух строк (или операция XOR) производится следующим образом: если соответствующие биты совпадают, то результат их сложения – ноль. Например, $0011 \text{ XOR } 0101 = 0110$.

Пример работы алгоритма: $K(0010) = 00101101$, если помеха во втором бите, то $D(01101101) = 0110 \text{ XOR } 1101 = 1011$. Декодировано 0010.

Задача 4 (Максимум – 25 баллов).

На военных учениях пушка выстреливает ядро с начальной скоростью $V_0 = 230$ м/с под углом $\alpha = 60^\circ$ к горизонту. Пушечное ядро пролетает в воздухе по баллистической траектории без трения и по задумке попадает в квадратную мишень, находящуюся на расстоянии L , верхний и нижний край которой расположены на высотах $h_2 = 3,5$ м и $h_1 = 1.5$ м соответственно.

Однако в силу несовершенности технических характеристик пушки присутствует погрешность, т.е. не всегда можно гарантировать точное попадание снаряда в мишень. Считайте, что ядро попадает внутрь круга радиусом $R = 1$ м, центр которого — это расчётная точка попадания снаряда, с вероятностью 100%. Вероятность попадания вне круга равна нулю. Размером ядра пренебречь.

1. Найдите, какие значения подойдут для L , чтобы ядро попадало в мишень с вероятностью более 50% (в метрах, округлив до десятых).
2. Каким должно быть L , чтобы снаряд попадал в мишень со 100% вероятностью (в метрах, округлив до десятых)?
3. С какой расчётной скоростью пробивает ядро мишень, если можно сказать, что оно точно попадёт в мишень (в м/с, округлив до десятых)?
4. До какой максимальной высоты поднялся снаряд (в метрах, округлив до десятых)?
5. При каком угле выстрела снаряд бы улетел на максимальное расстояние, если бы не было мишени (в градусах, округлив до целых)?

Задача 5 (Максимум – 25 баллов).

Рассматривается система автоматического управления межвагонной дверью поезда на базе микроконтроллера Arduino, предназначенная для управления открытием и закрытием двери по сигналу датчика силы нажатия.

На рисунке 2 представлена монтажная схема разработанного прототипа, она включает в себя:

- микроконтроллер Arduino Uno,
- делитель напряжения, состоящий из резисторов R1-R5 с известными сопротивлением $R1 = 24 \text{ кОм}$, $R2 = R3 = 6 \text{ кОм}$, $R4 = 2 \text{ кОм}$,
- идеальный мотор, корректно подключенный через драйвер мотора L293D и обеспечивающий вращение как по часовой стрелке, так и против.
- датчик силы нажатия, выполняющий роль кнопки управления и позволяющий измерять силу в диапазоне $[0; 10] \text{ Н}$.
- RGB-светодиод для индикации текущего состояния системы,
- монтажные платы и соединительные провода.

Проанализируйте монтажную схему и программный код (рисунки 3а,3б), и ответьте на следующие вопросы:

1. Перерисуйте монтажную схему делителя напряжения в виде электрической схемы цепи. Найдите сопротивление R5, если известно, что сопротивление всего делителя напряжения равняется $9,1 \text{ кОм}$.
2. Объясните работу программного кода построчно, поясните основную логику алгоритма работы микроконтроллера, представленных функций.
3. Какие цвета светодиода используются для индикации состояния системы? Запишите их RGB-коды и поясните как формируется цвет при использовании цветовой модели RGB.
4. Нарисуйте диаграмму состояний системы и опишите условия переходов между состояниями.
5. Какую минимальную сильную нужно приложить к датчику силы для инициации процесса открытия двери, если делитель напряжения в данной схеме обеспечивает максимальное значение $\text{fsrValue} = 829$? Учтите, что калибровочная характеристика датчика силы является линейной.

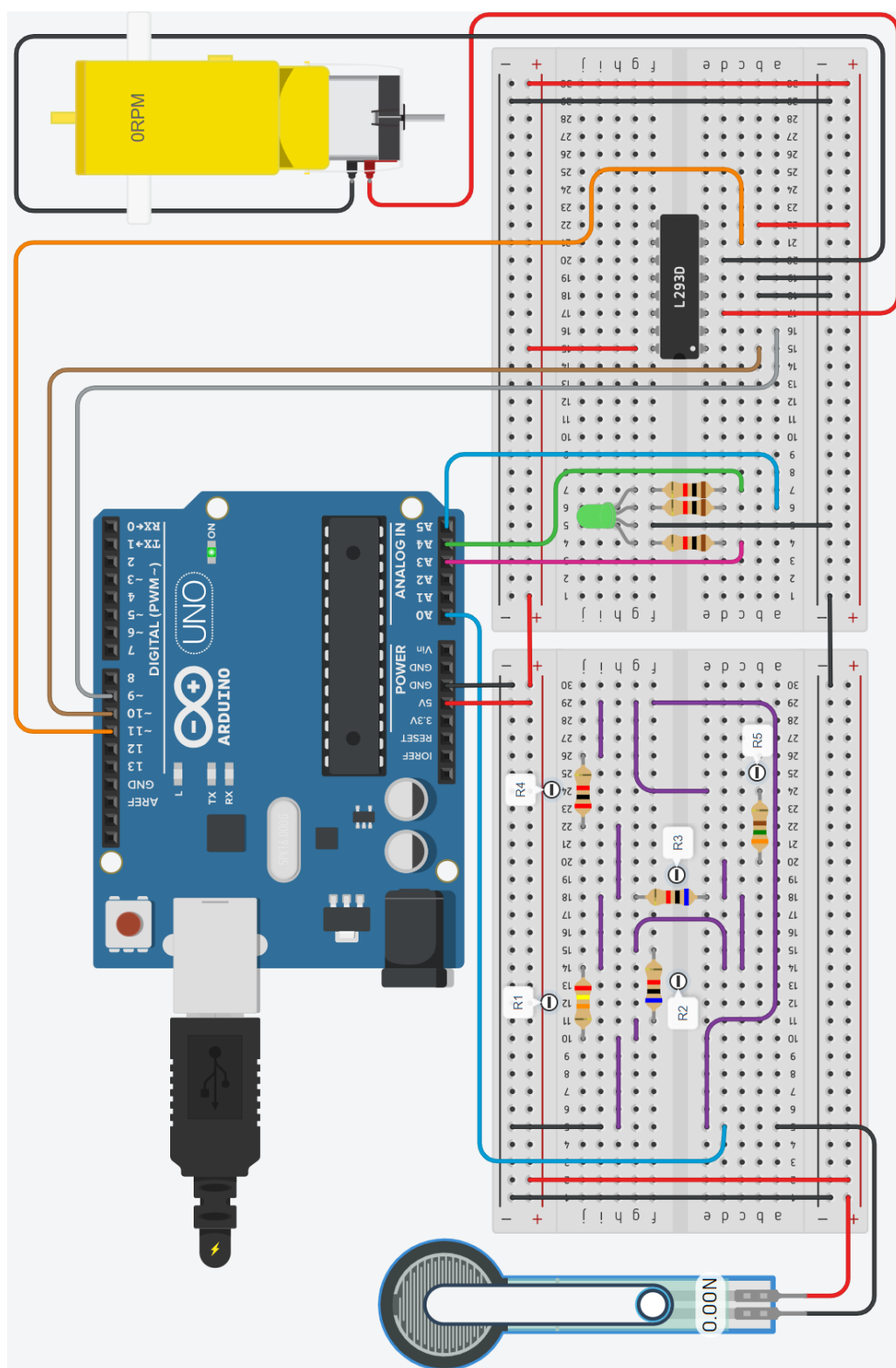


Рисунок 2. Монтажная схема

```
1  const int fsrPin    = A0;
2  const int ledRed    = A3;
3  const int ledGreen  = A4;
4  const int ledBlue   = A5;
5  const int motorPWM  = 10;
6  const int motorIN1  = 9;
7  const int motorIN2  = 11;
8  const int PRESS_THRESHOLD = 100;
9  const int HALF_POWER = 128;
10 const int FULL_POWER = 255;
11 const unsigned long MOTOR_TIME = 6000;
12 const unsigned long NO_PRESS_TIMEOUT = 4000;
13 enum MotorState {
14     IDLE,
15     OPENING,
16     WAIT_BEFORE_CLOSING,
17     CLOSING
18 };
19 MotorState state = IDLE;
20 unsigned long stateTimer = 0;
21 unsigned long lastPressTime = 0;
22
23 void motorStop() {
24     analogWrite(motorPWM, 0);
25     digitalWrite(motorIN1, LOW);
26     digitalWrite(motorIN2, LOW);
27 }
28 void motorOpen() {
29     digitalWrite(motorIN1, HIGH);
30     digitalWrite(motorIN2, LOW);
31     analogWrite(motorPWM, FULL_POWER);
32 }
33 void motorClose() {
34     digitalWrite(motorIN1, LOW);
35     digitalWrite(motorIN2, HIGH);
36     analogWrite(motorPWM, HALF_POWER);
37 }
38 void ledIdle() {
39     analogWrite(ledRed, 0);
40     analogWrite(ledGreen, 255);
41     analogWrite(ledBlue, 0);
42 }
43 void ledActive() {
44     analogWrite(ledRed, 240);
45     analogWrite(ledGreen, 240);
46     analogWrite(ledBlue, 240);
47 }
```

Рисунок 3а. Программный код, загруженный в память Arduino Uno.

```
48 void setup() {
49   pinMode(ledRed, OUTPUT);
50   pinMode(ledGreen, OUTPUT);
51   pinMode(ledBlue, OUTPUT);
52   pinMode(motorPWM, OUTPUT);
53   pinMode(motorIN1, OUTPUT);
54   pinMode(motorIN2, OUTPUT);
55   motorStop();
56   ledIdle();
57 }
58
59 void loop() {
60   unsigned long now = millis();
61   int fsrValue = analogRead(fsrPin);
62   switch (state) {
63     case IDLE:
64       ledIdle();
65       if (fsrValue > PRESS_THRESHOLD) {
66         ledActive();
67         lastPressTime = now;
68         motorOpen();
69         state = OPENING;
70         stateTimer = now;
71       }
72       break;
73     case OPENING:
74       if (now - stateTimer >= MOTOR_TIME) {
75         motorStop();
76         state = WAIT_BEFORE_CLOSING;
77         stateTimer = now;
78       }
79       break;
80     case WAIT_BEFORE_CLOSING:
81       if (fsrValue > PRESS_THRESHOLD) {
82         lastPressTime = now;
83       }
84       if (now - lastPressTime >= NO_PRESS_TIMEOUT) {
85         motorClose();
86         state = CLOSING;
87         stateTimer = now;
88       }
89       break;
90     case CLOSING:
91       if (now - stateTimer >= MOTOR_TIME) {
92         motorStop();
93         state = IDLE;
94       }
95       break;
96   }
97 }
```

Рисунок 3в. Программный код, загруженный в память Arduino Uno.