КОМПЛЕКТ ЗАДАНИЙ ДЛЯ 10–11 КЛАССОВ

Муниципального этапа всероссийской олимпиады школьников по труду (технологии) по профилю

«Техника, технологии и техническое творчество»

Автоматизированные технические системы

10–11 класс

Ответы

Разбор Этапа 1. Инженерный расчёт

Задача 1. Расчёт резистора
Дано: U_пит = 5 В, U_led = 2,0 В, I_led = 20 мА = 0,02 А
R = (5 − 2) / 0,02 = 3 / 0,02 = **150 Ом**
Ближайший стандартный номинал ряда E24 — **150 Ом** (есть в ряду).
$P = I^2 \times R = 0{,}02^2 \times 150 = 0{,}0004 \times 150 =$ **0,06 Вт**
Минимальная мощность резистора — 0,06 Вт. Стандарт — 0,125 Вт или 0,25 Вт.
Ответ: R = 150 Ом, P ≥ 0,06 Вт (рекомендуется 0,125 Вт).

Задача 2. Расчёт кинематики привода
Дано:
$n_{мотора}$=10 об/с=600 об/мин
Z1=20, Z2=80, d=0,2 м, π≈3,14
Передаточное отношение внешней передачи:
i=Z 2 /Z 1 =80/20=4.
Обороты колеса:
$n_{колеса} = n_{мотора}$ /i=600/4=150 об/мин.
Скорость робота:
$v=(\pi \times d \times n_{колеса})/60=(3{,}14\times 0{,}2\times 150)/60=1{,}57$ м/с.

Ответ: v=1,57 м/с.

Разбор Этапа 2. Эталонная 3D-модель
Эталонная модель содержит:
Платину шасси 180 × 120 × 3 мм со скруглёнными углами
Крепления: мотор-редукторы, HC-SR04 (спереди), Arduino (4 отв. M3), драйвер L298N (4 отв. M3).
2 ребра жёсткости высотой около 5 мм, шириной около 3 мм, расположенные симметрично

Сборка: шасси + моторы + датчик + Arduino + драйвер. Колёса и аккумулятор **НЕ** устанавливаются.
Фаски 1 × 45° на всех внешних кромках
Сборка содержит: шасси + 2 мотор-редуктора + датчик + плата Arduino, расположенные согласно габаритным чертежам.

Разбор Этапа 3. Эталонная документация
Рабочий чертёж пластины шасси:
Формат А3, рамка по ГОСТ 2.301, основная надпись по ГОСТ 2.104 (форма 1)
3 вида: главный (сверху), слева, сверху
Размеры: габаритные (180, 120, 3), присоединительные (межосевые расстояния отверстий), конструктивные (радиусы скруглений, размеры пазов)
Указание допусков и формы шероховатости
Материал: текст «Текстолит 3 ГОСТ» или «Пластик ABS 3»

Разбор Этапа 4. Пример скетча Arduino

```
// КОНСТАНТЫ ПИНОВ
const int PIN_TRIG = 9;        // Триггер ультразвукового датчика
const int PIN_ECHO = 10;       // Эхо ультразвукового датчика
const int PIN_LED  = 13;       // Светодиод индикации тревоги
const int PIN_MOTOR_A1 = 4;    // Драйвер мотора A, вход 1
const int PIN_MOTOR_A2 = 5;    // Драйвер мотора A, вход 2
const int PIN_MOTOR_B1 = 6;    // Драйвер мотора B, вход 1
const int PIN_MOTOR_B2 = 7;    // Драйвер мотора B, вход 2

// ПАРАМЕТРЫ
const int DISTANCE_THRESHOLD = 15;  // Порог срабатывания, см
const int BACK_DURATION_MS = 1000;  // Время отъезда назад, мс
const int TURN_DURATION_MS = 500;   // Время поворота, мс

// ИНИЦИАЛИЗАЦИЯ
void setup() {
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
  pinMode(PIN_LED, OUTPUT);
  pinMode(PIN_MOTOR_A1, OUTPUT);
  pinMode(PIN_MOTOR_A2, OUTPUT);
  pinMode(PIN_MOTOR_B1, OUTPUT);
  pinMode(PIN_MOTOR_B2, OUTPUT);
```

```
  digitalWrite(PIN_LED, LOW);  // Светодиод выключен
}

//ОСНОВНОЙ ЦИКЛ
void loop() {
  int distance = measureDistance();

  if (distance < DISTANCE_THRESHOLD) {
    // Обнаружено препятствие
    stopMotors();
    digitalWrite(PIN_LED, HIGH);     // Включаем индикацию тревоги
    delay(200);

    moveBackward(BACK_DURATION_MS);  // Отъезжаем назад
    stopMotors();

    turnRight(TURN_DURATION_MS);     // Поворачиваем направо
    stopMotors();

    digitalWrite(PIN_LED, LOW);      // Выключаем индикацию
  } else {
    moveForward();                   // Движемся вперёд
  }

  delay(50);  // Пауза между измерениями
}

//ФУНКЦИИ УПРАВЛЕНИЯ

// Измерение расстояния ультразвуковым датчиком
int measureDistance() {
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(PIN_TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);

  long duration = pulseIn(PIN_ECHO, HIGH);
  int distance = duration * 0.034 / 2;  // Перевод в сантиметры
  return distance;
}
```

```
// Движение вперёд
void moveForward() {
  digitalWrite(PIN_MOTOR_A1, HIGH);
  digitalWrite(PIN_MOTOR_A2, LOW);
  digitalWrite(PIN_MOTOR_B1, HIGH);
  digitalWrite(PIN_MOTOR_B2, LOW);
}

// Движение назад
void moveBackward(int durationMs) {
  digitalWrite(PIN_MOTOR_A1, LOW);
  digitalWrite(PIN_MOTOR_A2, HIGH);
  digitalWrite(PIN_MOTOR_B1, LOW);
  digitalWrite(PIN_MOTOR_B2, HIGH);
  delay(durationMs);
}

// Поворот направо (один мотор вперёд, другой назад)
void turnRight(int durationMs) {
  digitalWrite(PIN_MOTOR_A1, HIGH);
  digitalWrite(PIN_MOTOR_A2, LOW);
  digitalWrite(PIN_MOTOR_B1, LOW);
  digitalWrite(PIN_MOTOR_B2, HIGH);
  delay(durationMs);
}

// Остановка двигателей
void stopMotors() {
  digitalWrite(PIN_MOTOR_A1, LOW);
  digitalWrite(PIN_MOTOR_A2, LOW);
  digitalWrite(PIN_MOTOR_B1, LOW);
  digitalWrite(PIN_MOTOR_B2, LOW);
}
```