

№ стола _____

Название команды: _____

ОЦЕНКА ФИНАЛЬНОГО ПРАКТИЧЕСКОГО КОМАНДНОГО ЗАДАНИЯ

0.Фотографии устройства	Да/Нет (+/-)	Балл
0.1. Выполнено фотографирование устройства, фотографии загружены в облачную папку.		5
1.3D-моделирование (выбор <u>одного</u> верного утверждения)	Да/Нет (+/-)	Балл
1.1.3D-модели сбрасывателя в САПР представлена (есть файлы в требуемом формате, в облачной папке), присутствуют грубые ошибки.		15
1.2. 3D-модели сбрасывателя в САПР представлена (есть файлы в требуемом формате, в облачной папке). 3D-модели сбрасывателя в САПР не содержит ошибок.		30
2. Электрическая принципиальная схема (выбор <u>одного</u> верного утверждения)	Да/Нет (+/-)	Балл
2.1. Представлена и загружена в папку электрическая принципиальная схема с серьезными ошибками.		10
2.2. Представлена и загружена в папку электрическая принципиальная схема с незначительными ошибками.		20
2.3. Представлена и загружена в папку электрическая принципиальная схема без ошибок.		30
3. Сборка (выбор <u>одного</u> верного утверждения)	Да/Нет (+/-)	Балл
3.1. Выполнена сборка конструкции согласно чертежам с серьезными ошибками. Устройство не имеет возможность функционировать согласно ТЗ. Фотографии результатов сборки загружены в папку.		10
3.2. Выполнена сборка конструкции согласно чертежам с небольшими ошибками. Устройство имеет возможность функционировать согласно ТЗ частично. Фотографии результатов сборки загружены в папку.		20
3.3. Выполнена сборка конструкции согласно чертежам без ошибок. Устройство имеет возможность функционировать согласно ТЗ. Фотографии результатов сборки загружены в папку.		30

4. Монтаж электронных компонентов (выбор <u>одного</u> верного утверждения)	Да/Нет (+/-)	Балл
4.1. Выполнен монтаж компонентов согласно схеме с серьезными ошибками. Электроника не функционирует (если электроника выведена из строя участникам выставляется 0 баллов).		10
4.2. Выполнен монтаж компонентов согласно схеме с небольшими ошибками (не все элементы подключены, есть ошибки соединения, не прямые соединения проводки). Электроника функционирует частично.		20
4.3. Выполнен монтаж компонентов согласно схеме без ошибок (прямые соединения проводки, цветные провода, все элементы предложенной схемы задействованы).		30
5. Выполнение испытания №2 (отметка напротив <u>КАЖДОГО</u> утверждения)	Да/Нет (+/-)	Балл
5.1. Реализована возможность выбора режима работы ПАК (выбор режима А, режима В, возврат к выбору режима “#”).		10
5.2. Реализован функционал взвешивания объекта и вывода информации о весе объекта на LCD-дисплей.		15
5.3. Реализован функционал калибровки весов – погрешность измерения веса объекта менее 2 грамм (в противном случае выставляется – 0 баллов).		15
5.4. Реализована возможность внесения информации о цене за 100 граммов груза в ПАК через клавиатуру, производится автоматический расчет итоговой стоимости объекта с выводом информации на дисплей.		15
5.5. Реализован функционал расчета стоимости объекта и вывода информации о стоимости объекта на LCD-дисплей.		15
5.6. Реализован функционал для сохранения результатов взвешивания и расчета стоимости (оценивается сохранение каждого отдельного результата работы – по 20 баллов максимум).		20
6. Выполнение испытания №3 (отметка напротив <u>КАЖДОГО</u> утверждения)	Да/Нет (+/-)	Балл
6.1. Реализована возможность выбора режима работы ПАК (выбор режима С, режима D, возврат к выбору режима “#”).		10
6.2. Реализован функционал внесения информации об эталонном значении веса для сортировки с использованием клавиатуры.		10

6.3. Реализован функционал сортировки объектов по весу, объект перемещен в нужную зону.		15
6.4. Реализован функционал последовательного внесения информации о стоимости за 100 граммов груза и эталонной стоимости груза для сортировки.		15
6.5. Реализован функционал сортировки объектов по стоимости, объект перемещен в нужную зону.		15
6.6. Реализован функционал для сохранения результатов сортировки (оценивается сохранение каждого отдельного результата работы – по 20 баллов максимум).		20
8. Штрафы	Да/Нет (+/-)	Балл
8.1. В случае очевидного вывода из строя электронных компонентов или элементов конструкции участниками команды к итоговому баллу применяется штраф		- 100
ИТОГОВЫЙ БАЛЛ (максимальный балл – 300 минимальный балл – 0)		

Эксперт: (ФИО, подпись) _____

Инструкция для эксперта по проведению испытаний

Испытание №1:

1. Проверяется наличие документов и загрузки в облачную папку (фото, 3д модели, электрическая принципиальная схема).
2. Проверяется корректность загруженных файлов.
3. Визуально проверяется сборка (конструкция и монтаж).

Испытание №2:

1. Проверяется система управления с помощью клавиатуры.
2. Проверяется взвешивание объектов – взвешивается объект 50 граммов, 25 граммов, и оба объекта 75 граммов. На LCD-дисплее должна быть выведена информация ТОЛЬКО о весе.
3. Проверяется точность измерения веса – должна быть ± 2 грамма.
4. После измерения веса переход к выбору другого режима работы должен быть реализован через нажатие “#”.
5. Для проверки расчета стоимости выбирается **объект 25 граммов, стоимость за 100 граммов – 125**. Итоговая стоимость объекта для вывода на дисплей: **31.25 ± 2.5**
6. Проверяется функция сохранения в файл. Файл должен выглядеть следующим образом:

1, A, 25, None, None
2, A, 50, None, None
3, A, 100, None, None
4, B, 25, 31.25 +- 2.5, None

Испытание №3:

1. Проверяется система управления с помощью клавиатуры.
2. Проверяется сортировка объектов по весу. Взвешивается объект **50 граммов, эталонное значение веса – 75 грамм**.
3. **Результат сортировки по весу: зона №1.**
4. Проверяется сортировка объектов по стоимости. Взвешивается объект: **25 граммов**, значение стоимости за 100 граммов: **200**, эталонное значение стоимости – **40**.
5. **Результат сортировки по стоимости: зона №2.**
6. После измерения веса переход к выбору другого режима работы должен быть реализован через нажатие “#”.
7. Проверяется функция сохранения в файл. Файл должен выглядеть следующим образом:

5, C, 50, None, zone 1
6. D, 25, 50.00 +- 2.5, zone 2

Весы для сортировки

Эталонные решения задач в рамках выполнения командного практического задания финала (поскольку задание является творческим, правильные результаты выполнения могут быть иметь разное представление).

Видео, демонстрирующее правильную работу разработанного ПАК:
<https://disk.yandex.ru/d/HEQjYVvhfre7ug>

Задачи, которые необходимо выполнить участникам команды в рамках выполнения командного практического задания финала.

Подготовка включает 5 основных задач (рекомендуется выполнять задачи параллельно с разделением ответственности между участниками команды):

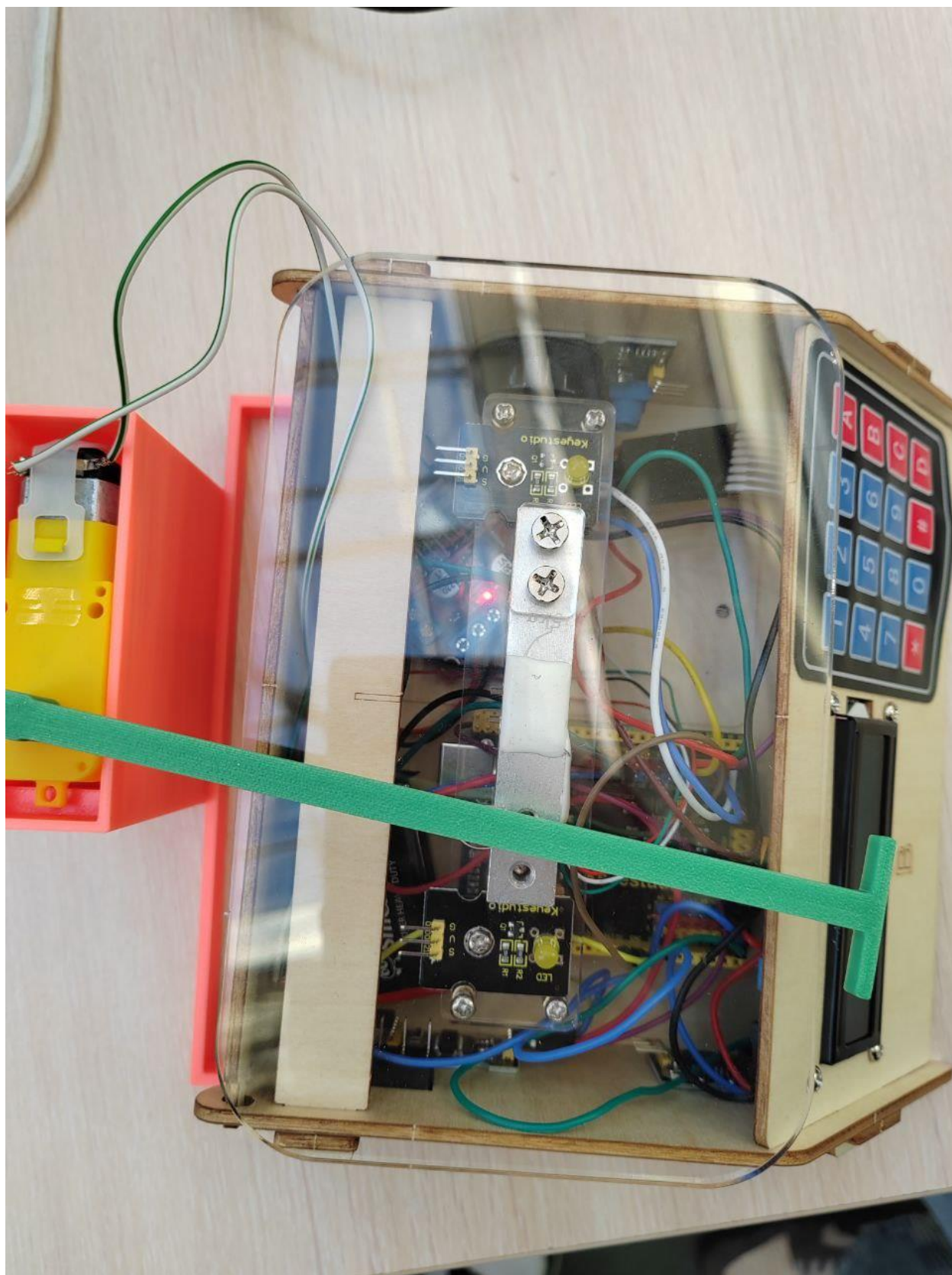
1. Разработать ПАК в соответствии с требованиями пункта II задания, способный проходить испытания в соответствии с регламентом испытаний (пункт V).

Должна быть выполнена сборка конструкции согласно чертежам без ошибок и должен быть выполнен монтаж компонентов согласно монтажной схеме без ошибок (прямые соединения проводки, цветные провода, все элементы предложенной схемы задействованы). Устройство должно иметь возможность функционировать согласно регламенту испытания ТЗ. Фотографии результатов сборки загружены в папку.

В случае обнаружения небольших ошибок или неточностей – выставляется частичный балл. В отсутствия результатов работы по данному пункту – выставляется 0 баллов.

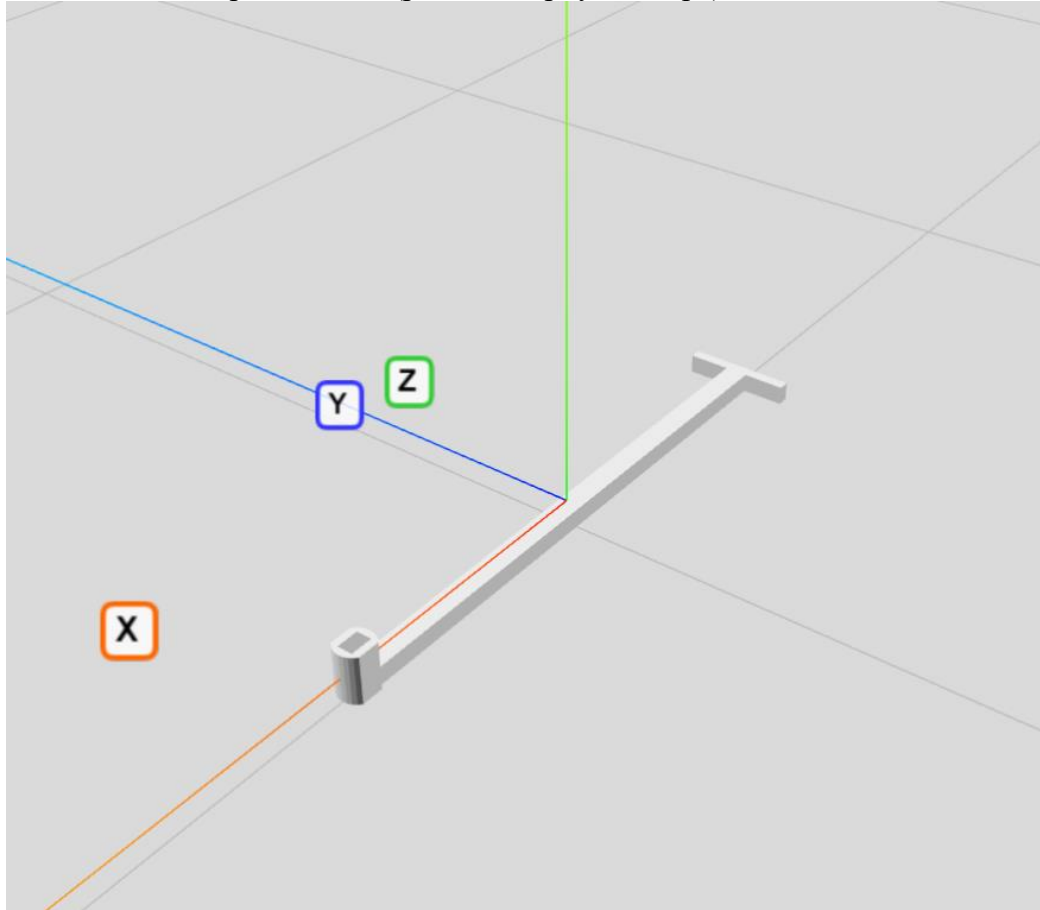
Результат, оценивающийся на полный балл:

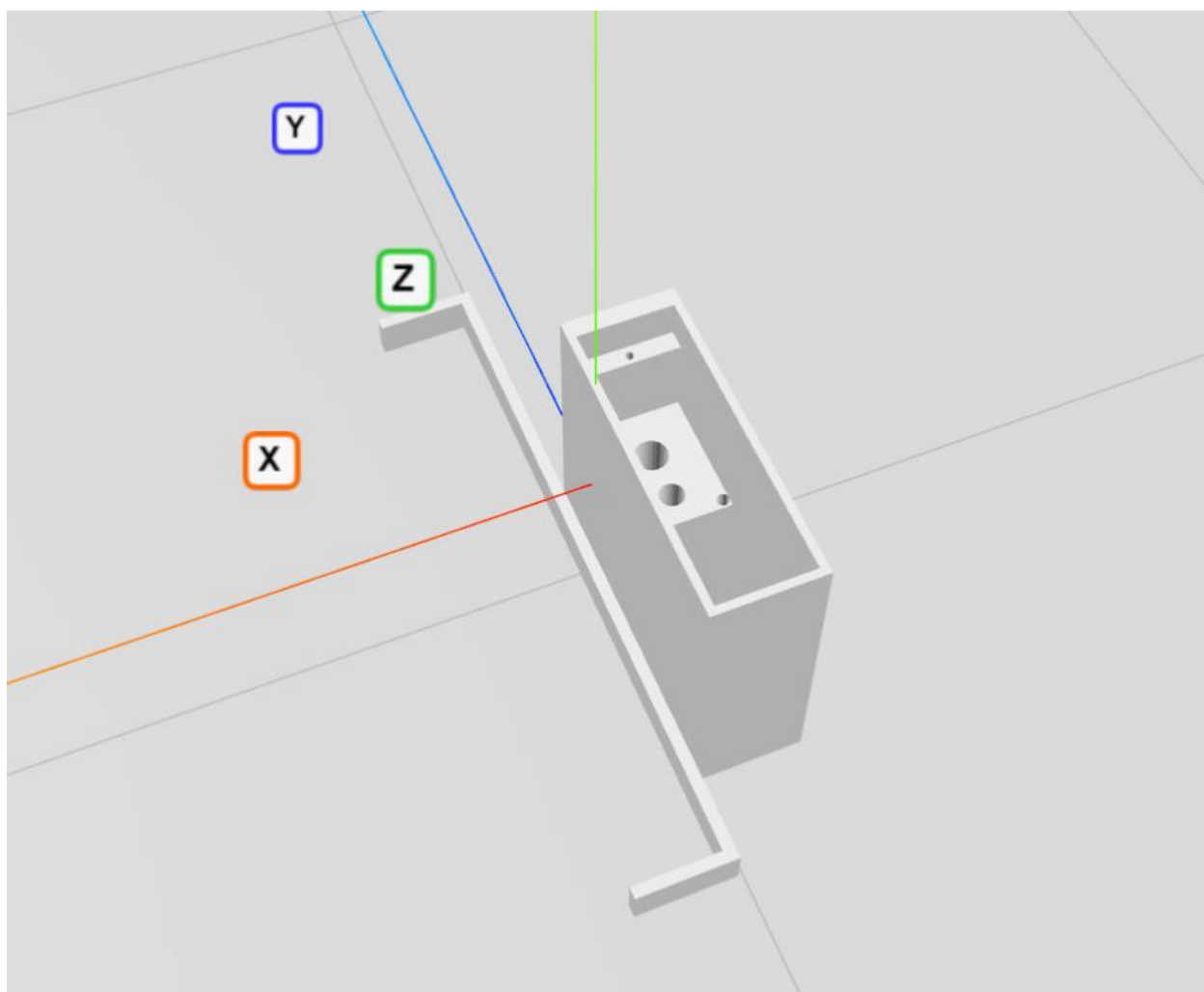




2. С использованием предоставленных 3D моделей, сборочных чертежей и чертежей элементов необходимо разработать:

- 3D модели сбрасывателя (рейка и корпус мотора):



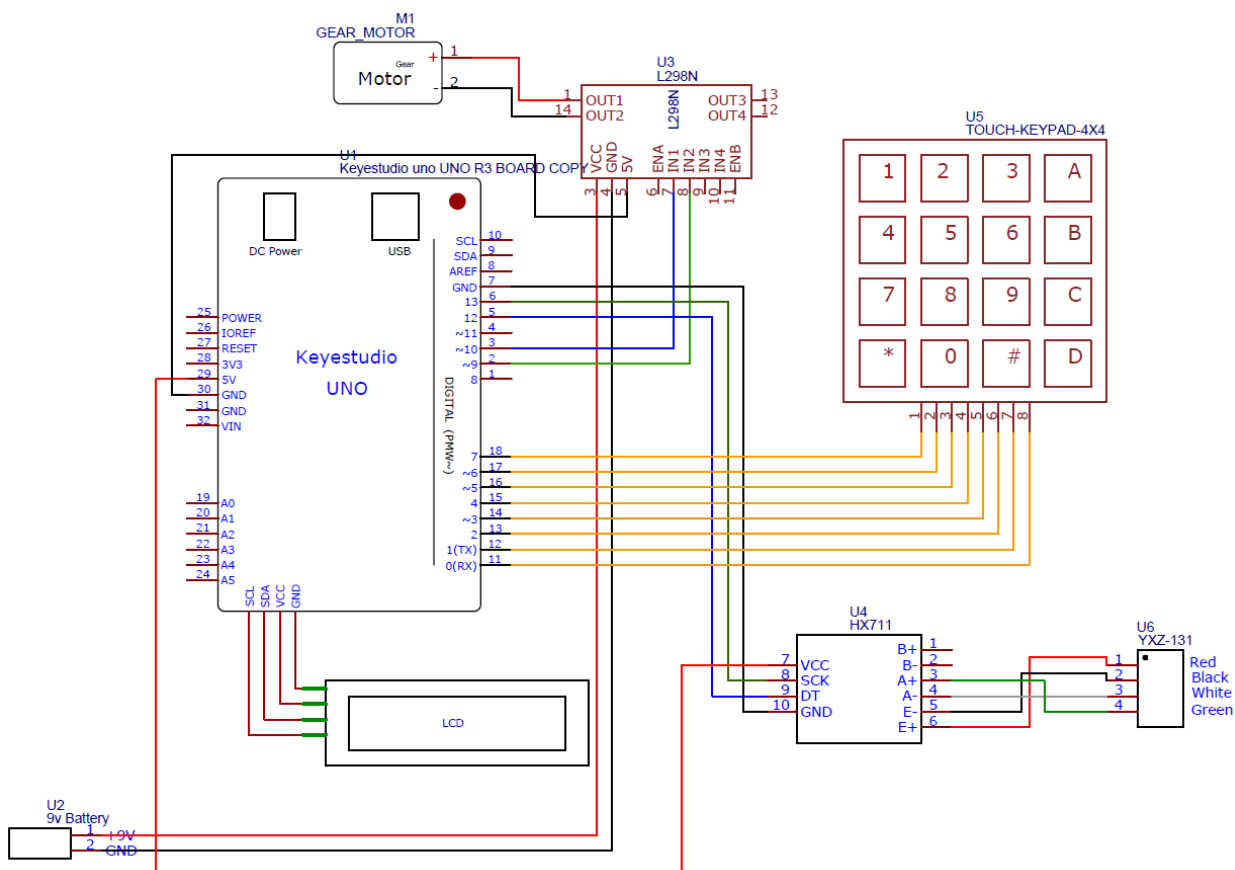


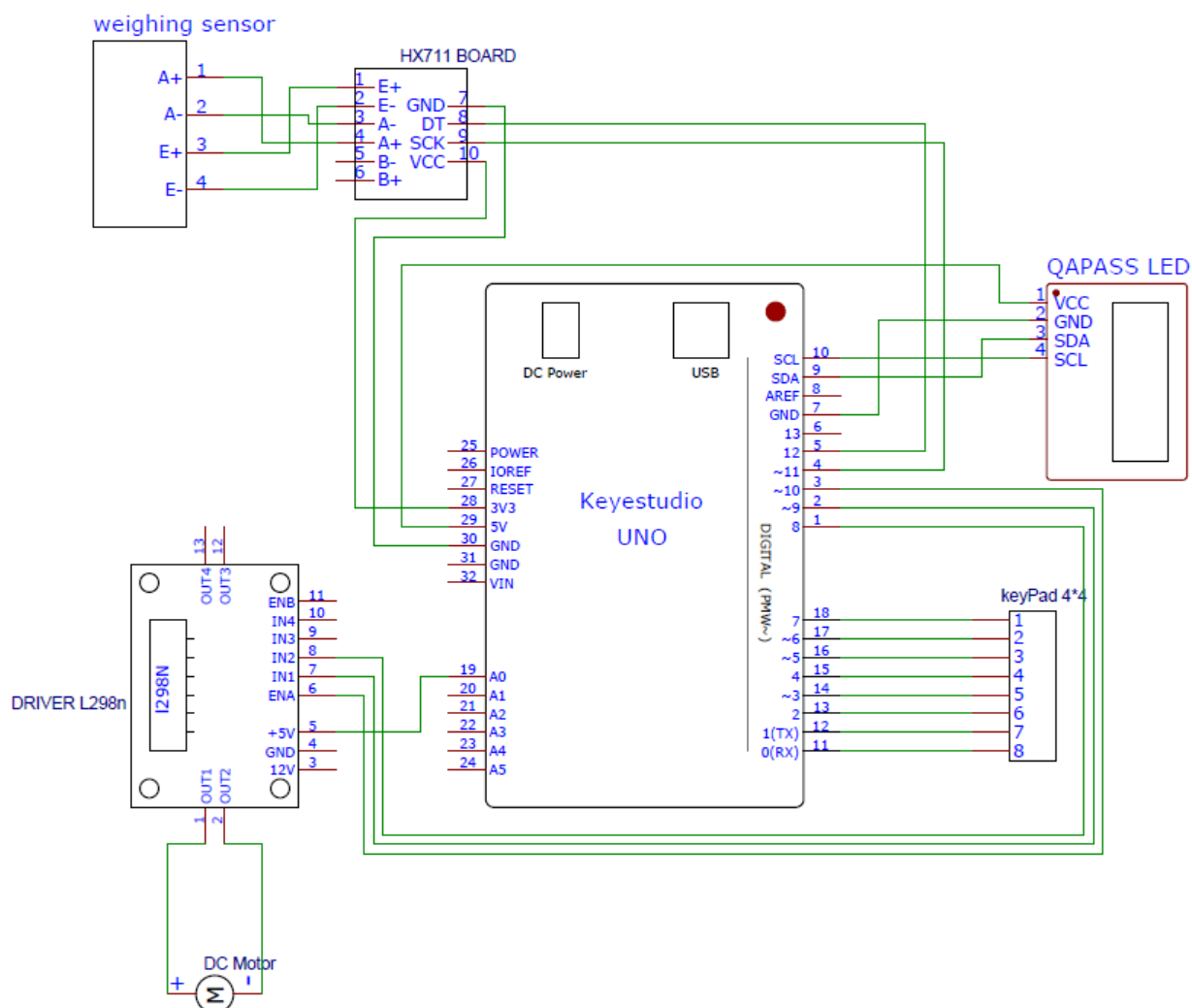
3D-модели в САПР не должны содержать ошибки.

В случае обнаружения небольших ошибок или неточностей – выставляется частичный балл.

В случае отсутствия результатов работы по данному пункту – выставляется 0 баллов.

3. С использованием монтажной схемы разработать электрическую принципиальную схему устройства с использованием в Fritzing или EasyEDA (необходимо использовать все электронные компоненты из предложенного набора).
Результат, оценивающийся на полный балл, формат .pdf:





Полный балл выставляется в случае, если представлена и загружена в папку электрическая принципиальная схема без ошибок. Электрическая принципиальная схема должна соответствовать выданной монтажной схеме, включать в себя все необходимые электронные компоненты и быть верно реализованной с точки зрения теории электрических цепей.

В случае обнаружения небольших ошибок или неточностей – выставляется частичный балл. В отсутствия результатов работы по данному пункту – выставляется 0 баллов.

4. Разработать программное обеспечение для управления ПАК и процедурой сканирования области с использованием ArduinoIDE (допускается использование любых готовых библиотек).

Результат, оценивающийся на полный балл:

Код микроконтроллера:

```
// Код для микроконтроллера устройства

/*
  keystudio Electronic scale
  Extended Functionality: Weighing, Pricing, Sorting, Logging
*/

#include <Keypad.h>
#include <HX711.h>
#include <EEPROM.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
HX711 hx(12, 13);

const byte ROWS = 4, COLS = 4;
char keyMap[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {2, 3, 4, 5};
byte colPins[COLS] = {6, 7, 8, 9};
Keypad myKeypad = Keypad(makeKeymap(keyMap), rowPins, colPins, ROWS, COLS);

double ratio, offset;
int experimentID = 1;
int motorPin1 = 10, motorPin2 = 11;

void setup() {
  Serial.begin(9600);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  lcd.init(); lcd.backlight();
  EEPROM.get(0, ratio);
  offset = hx.tare();
  hx.set_offset(offset);
}
```

```
lcd.print("Welcome! HX711");
delay(2000);
lcd.clear();
showMainMenu();
}

void loop() {
  char key = myKeypad.getKey();
  if (key) handleKeyPress(key);
}

void handleKeyPress(char key) {
  switch (key) {
    case 'A': modeWeighing(); break;
    case 'B': modePricing(); break;
    case 'C': modeSortWeight(); break;
    case 'D': modeSortPrice(); break;
    case '*': calibrate(); break;
  }
}

void modeWeighing() {
  double weight;
  while (true) {
    weight = getWeight();
    lcd.setCursor(0, 0);
    lcd.print("WEIGHT: " + String(weight, 2) + " g ");
    if (myKeypad.getKey() == '#') {
      logData(experimentID++, 'A', weight, "None", "None");
      showMainMenu();
      return;
    }
  }
}

void modePricing() {
  double pricePer100g = getUserInput("Enter price/100g");
  double weight;
  while (true) {
    weight = getWeight();
    double price = (weight / 100) * pricePer100g;
    lcd.setCursor(0, 0);
    lcd.print("WEIGHT: " + String(weight, 2) + " g ");
    lcd.setCursor(0, 1);
    lcd.print("PRICE: " + String(price, 2));
    if (myKeypad.getKey() == '#') {
      logData(experimentID++, 'B', weight, String(price, 2), "None");
      showMainMenu();
      return;
    }
  }
}
```

```
void modeSortWeight() {
    double threshold = getUserInput("Enter weight lim");
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Put weight");
    while (true) {
        double weight = getWeight();
        if (weight > 2.) {
            delay(300);
            weight = getWeight();
            String zone = (weight >= threshold) ? "zone 1" : "zone 2";
            moveMotor(zone);
            logData(experimentID++, 'C', weight, "None", zone);
            showMainMenu();
            return;
        }
    }
}

void modeSortPrice() {
    double pricePer100g = getUserInput("Enter price/100g");
    double priceThreshold = getUserInput("Enter price lim");
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Put weight");
    while (true) {
        double weight = getWeight();
        if (weight > 2.) {
            delay(300);
            weight = getWeight();
            double price = (weight / 100) * pricePer100g;
            String zone = (price >= priceThreshold) ? "zone 1" : "zone 2";
            moveMotor(zone);
            logData(experimentID++, 'D', weight, String(price, 2), zone);
            showMainMenu();
            return;
        }
    }
}

void moveMotor(String zone) {
    if (zone == "zone 1") analogWrite(motorPin1, 80);
    else analogWrite(motorPin2, 80);
    delay(1600);
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 0);
}

double getWeight() {
    double weight = hx.bias_read() * ratio;
    if (abs(weight) < 1.5) weight = 0.;
}
```

```
    return weight;
}

double getUserInput(String prompt) {
    lcd.clear();
    lcd.print(prompt);
    String input = "";
    while (true) {
        char key = myKeypad.getKey();
        if (key == '#') return input.toDouble();
        if (key && isDigit(key)) {
            input += key;
            lcd.setCursor(0, 1);
            lcd.print(input);
        }
        delay(10);
    }
}

void logData(int id, char mode, double weight, String price, String zone) {
    Serial.println(String(id) + ", " + mode + ", " + String(weight, 2) + ", " + price + ", " + zone);
}

void showMainMenu() {
    lcd.clear();
    lcd.print("Select mode: A-D");
}

void calibrate() {
    lcd.clear();
    double sum = 0;
    for (int i = 0; i < 10; i++) {
        sum += hx.bias_read();
    }
    sum = sum / 10;
    lcd.clear();
    lcd.print("PUT ON 50g Farmar");
    lcd.setCursor(0, 1);
    lcd.print("9 seconds later");
    int countdown = 9;
    while (countdown != 0) {
        lcd.setCursor(0, 1);
        lcd.print(countdown);
        countdown--;
        delay(1000);
    }
    double sum1 = 0;
    for (int i = 0; i < 10; i++) {
        sum1 += hx.bias_read();
    }
    sum1 = sum1 / 10;
    ratio = abs(sum1 - sum);
}
```



```
ratio = 50 / ratio;
EEPROM.put(0, ratio); // save coefficient which needs compiling via above 1.6.0 IDE
lcd.clear();
lcd.print("CAL OK!");
delay(1000);
showMainMenu();
}
```

Код для логгирования результатов работы устройства на Python

```
import serial

def read_serial_log(port="/dev/tty.usbmodem14101", baudrate=9600, logfile="log.txt"):
    ser = serial.Serial(port, baudrate, timeout=1)
    first_entry = True

    while True:
        line = ser.readline().decode('utf-8').strip()
        if line:
            columns = line.split(", ")
            if columns[0] == "1":
                mode = "w" # Перезаписываем файл при новом эксперименте
                first_entry = False
            else:
                mode = "a" # Добавляем в конец файла

            with open(logfile, mode) as file:
                file.write(line + "\n")
            print("Logged:", line) # Для отладки

if __name__ == "__main__":
    read_serial_log()
```

Для оценки выполнения функциональных требований к разрабатываемому устройству и его программному обеспечению используется бинарная оценка – выполнено или не выполнено. Таким образом, при полном выполнении выставляется полный балл, при частичном выполнении или отсутствии результатов решения задачи выставляется 0 баллов.

5. Необходимо произвести фотографирование разработанного ПАК (не менее 3 фотографий с различных ракурсов).

Результаты, отображенные на непосредственно фотографиях, не оцениваются. Данная задача служит для фиксации наличия результатов работы участников.

