

Проект по технологии

Бортовая электроника ракеты команды «Пепелац»

v. 1.1

(испытано на пусковом комплексе космодрома «Восточный»)

Проект разработал:

Тарасов Степан,

Ученик 9Б класса

Наставники:

Черёмухин П.С.

Учитель технологии МОУ
«Инженерная школа города
Комсомольска-на-Амуре»

Пашин С.В.

Учитель физики МОУ
«Инженерная школа города
Комсомольска-на-Амуре»

г. Комсомольск-на-Амуре

2022

Содержание:

1. Актуальность, задачи и историческая справка по теме проекта
 - 1.1. Обоснование проблемы и формулировка темы проекта
 - 1.2. Историческая справка
2. Анализ прототипов
3. Анализ возможных идей. Выбор оптимальной идеи
4. Художественное проектирование: разработка концепции проекта и его значимость, создание эскизов
 - 4.1. Требования к изделию
 - 4.2. Определение метода или приёмов дизайн-проектирования
5. Обоснование и подбор материалов, разработка конструкторской
 - 5.1. Выбор материалов
6. Выбор технологии изготовления изделия. Технологическое описание изготовления изделия
 - 6.1. Технологическая карта
 - 6.2. Описание изготовления изделия
7. Оригинальность предложенных технико-технологических, инженерных или эргономических решений, новизна проекта
8. Экономическая и экологическая оценка будущего изделия и технологии его изготовления
 - 8.1. Расчет затрат на электронную составляющую проекта
 - 8.2. Расчет затрат на изготовление корпуса
 - 8.3. Расчет энергозатрат при производстве
 - 8.4. Расчет трудозатрат при производстве
 - 8.5. Экологическая оценка изделия
9. Рекламные предложения и перспективы внедрения изделия
10. Алгоритм работы с изделием
11. Список использованной литературы
12. Приложение А. Скetch (программная часть бортовой электроники)

1. Актуальность, задачи и историческая справка по теме проекта

1.1. Обоснование проблемы и формулировка темы проекта

На сегодняшний день авиакомпании совершают десятки или сотни полетов в день. Каждый из них зависит от погодных условий, которые на высоте могут отличаться от наземных. Пилотам и диспетчерам связи необходимо иметь представление о погодных условиях, но как их измерить? Существует очень мало способов измерить температуру, давление, влажность и другие параметры в воздухе. Один из них — запустить ракету.

Цель — создать прототип электроники ракеты, которая может брать показания воздушной среды.

Задача — сделать его максимально легким, эргономичным и функциональным.

1.2. Историческая справка

Ракетомоделизм — вид технического творчества, построение моделей ракет.

Запуски ракет можно проводить самостоятельно, организованного на спортивных соревнованиях. Обязательным условием является соблюдение правил техники безопасности.

Модель ракеты 3022 Payloader II после сборки, готова к пуску: головной обтекатель показан снизу, в котором имеется место для полезной нагрузки; обтекатель соединён с хвостовой частью и снабжён парашютом; все части собираются вместе перед пуском, а после отработки двигателя срабатывает вышибной заряд, ракета в воздухе раскрывается как на рисунке и опускается на парашюте.

Простые модели строятся из лёгких материалов (картон, бальза) и используют одноразовые, твердотопливные двигатели.

Основные компоненты ракеты: головной обтекатель, корпус, направляющие кольца, стабилизаторы, двигатель и тормозная система (парашют или тормозная лента).

В качестве полезной нагрузки могут использоваться: миниатюрный альтиметр, фотоаппарат, видеокамера и другие устройства для проведения экспериментов и обучения.

Основные характеристики двигателей: тяга-время. Все двигатели подразделяются на классы в зависимости от суммарного импульса.

Двигатели промышленного производства имеют маркировку на корпусе, численные значения параметров указаны в документации. Назначение пусковой установки — обеспечить вертикальное движение ракеты, пока не

будет достигнута скорость стабильного полёта. С помощью направляющих колец ракета крепится на направляющий стержень перед стартом.

Основные компоненты пусковой установки: стартовая плита, направляющий штырь, пульт управления, провода для подачи электропитания. Типичная проблема после запуска.

Основные пункты техники безопасности принятые Национальной ассоциацией ракетомоделистов:

- Использовать только лёгкие (не металлические) материалы для головного обтекателя, корпуса, стабилизаторов.
- Использовать только сертифицированные, не изменённые двигатели.
- Использовать электрическую систему запуска и зажигания.
- В случае неудачного старта — не приближаться к модели раньше, чем через 1 минуту.
- Во время старта находиться на безопасном расстоянии:
 - 4,5 метра для двигателей класса D и менее мощных
 - 9 метров для двигателей мощнее, чем класс D
- Ракета не должна весить больше 1500 грамм и не должна содержать более 125 граммов топлива.
- Не запускать ракету в какие-либо цели, облака или вблизи от самолётов и не размещать на ракете горючие или взрывчатые вещества.
- Не пытаться достать ракету с линий электропередач, высоких деревьев и других опасных мест.

2. Анализ Прототипов

Прототипом данной версии является предыдущая версия этой же электроники. v. 1.0. имела ряд недостатков:

- Отсутствие датчиков как таковых
- Обратная связь, которая предоставляла лишь высоту полета

В данной же версии недостатки были исправлены, улучшен корпус, облегченная конструкция, более надежная фиксация, а так же добавлена возможность получения температуры.

3. Анализ возможных идей. Выбор оптимальной идеи

Сейчас нужно определиться с необходимыми компонентами схемы. Первое — необходимо знать высоту. Второе — необходимо как-то сохранять информацию с высоты. Третье — нужно передать уже полученную информацию, если ракету не удалось найти. Четвертое — получение температуры на определенной высоте.

Варианты высотомера:

Есть один вариант — использовать барометр в качестве высотомера. Он, получая информацию об атмосферном давлении, посылает ее в контроллер, который по формуле высчитывает разность давлений и находит высоту, на которой находится ракета.

Варианты базы данных:

В качестве накопителя идеально подойдет SD-карта. Она занимает мало места, имеет большой объём и позволяет записывать на нее данные в удобном файле.

Варианты передатчика:

Необходимо было как-то передавать информацию на землю, чтобы получить хоть что-то, если ракета была утеряна. Оптимальным вариантом будет передавать информацию на землю радиосигналом. Чтобы информация была понятна, было принято решение закодировать все данные азбукой Морзе.

Варианты термометра:

В качестве термометра будет выступать уже упомянутый барометр BMP180, имеющий встроенный датчик температуры.

Итог: в схему войдут барометр-термометр, передатчик и SD-карта.

4. Художественное проектирование: разработка концепции проекта и его значимость, создание эскизов

4.1. Требования к изделию

Набор основных и дополнительных функций определяет требования к изделию на этапе художественного проектирования. Гармоничное сочетание основных и дополнительных функций должно сделать изделие универсальным.

- Функциональность
- Интерактивность
- Мобильность
- Надежность
- Энергоэффективность
- Низкая себестоимость
- Точность измерений
- Простота изготовления и управления

- Функциональность должна обеспечить получение высоты и давления, которые сохраняются на SD-карту и передаются по радиации на землю.
- Интерактивность заключается в постоянном получении информации с воздуха.
- Мобильность обуславливается установкой почти в любую ракету, так как корпус электроники можно редактировать.
- Надежность должна обеспечивать пожаробезопасность, стабильность в работе и исключение возможности отказа устройства.
- Энергоэффективность определяет время автономной работы, которое очень важно при проведении измерений.
- Низкая себестоимость относительно средних затрат на всю ракету.
- Точность измерений должна быть обеспечена качеством датчиков.
- Простота изготовления и управления позволит запустить мое изделие в серийное производство.

4.2. Определение метода или приёмов дизайн-проектирования

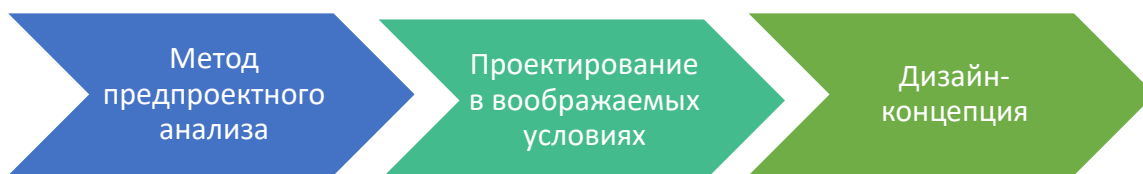


Рисунок 1. Методы дизайн-проектирования

В основу работы над проектом был положен метод предпроектного анализа, разделение процесса исследования предполагаемой ситуации на ряд этапов, самостоятельных по целям и результатам работы в разрезе самостоятельных направлений проектирования (**разработка электронной части, программный компонент, моделирование корпуса**).

1. Обследование, знакомство с ситуацией, контекстом размещения будущего объекта, перечнем свойств, которыми он должен обладать. Техника этого этапа: изучение аналогов, обзор литературных данных и реальных прототипов, выяснение их положительных и отрицательных качеств, формулировка прямых задач дальнейшей работы.

2. Постановка задания как проблемы, т.е. столкновения противоречий между обстоятельствами будущей жизни объекта и эксплуатационными характеристиками его структур.

3. Сравнение предложений, рассматривающие отдельные узлы проблемы, сведение их в разные варианты общего решения, и выбор среди этих вариантов наиболее эффективного.

Следующий метод — «проектирование в воображаемых условиях», когда реальные обстоятельства работы объекта условно подменяются неожиданными решениями («пробежка с электроникой по лестнице», «экстренный спуск» и др.)

Результатом предпроектного анализа и проектирования в воображаемых условиях является дизайн-концепция, принципиальная дизайнерская идея будущего проекта, но уже содержащая его реально представимые формы: инженерно-технические, пространственные, процессуальные.

Через проблематизацию проектной ситуации (описание противоречий проектного задания), а затем тематизацию (отбор «тем» - возможных вариантов ее решения), складывается целостная модель будущего объекта, реализуемая в актах собственно проектной работы — формообразования. А формообразование на концептуальной основе выводит художественное проектирование на уровень стилиобразования в рамках данной культуры, вырабатывая подходы к сознательному использованию признаков стиля.

5. Основание и подбор материалов, разработка конструкторской документации.

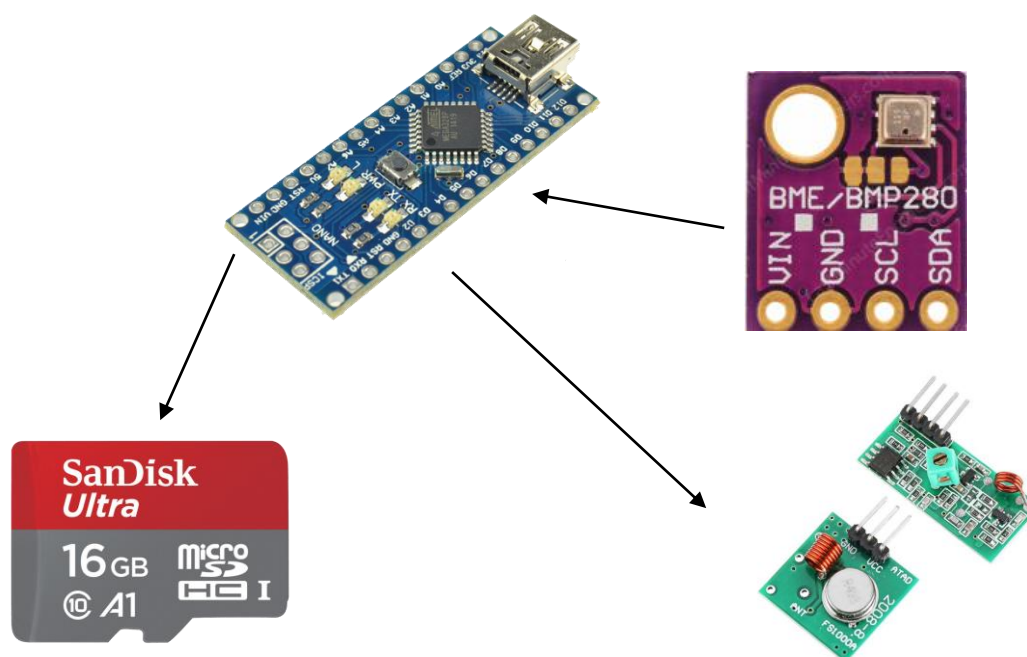


Рисунок 2. Функциональная схема компонентов

Функциональная схема устройства требует очень компактный корпус, который должен будет влезть в тонкую трубу. Корпус должен быть легким, чтобы не перетягивать центр тяжести ракеты к обтекателю. Детали должны быть качественными, чтобы не было нареканий при работе на большой высоте, где электроника в опасности. С учётом вышеназванных параметров производился выбор материалов, который обуславливает дальнейшее создание конструкторской документации и выбор технологии изготовления.

5.1. Выбор материалов

Таблица 1. Выбор материалов

	Фанера	Пластик ABS
Фото		
Плюсы	Прочность, возможность лазерной резки, фрезеровки и финишной обработки по всей площади.	Экологичный, легкий, возможность напечатать любую форму, низкая себестоимость, возможность вторичной переработки.
Минусы	Сложность придания нужной формы, тяжелый материал, много отходов.	Необходимо оборудование + создание 3D-модели

Вывод: использование 3D-печати будет лучшим решением, так как он легкий и корпус можно сделать любой формы, максимально выполнив все условия: малый вес, особая форма, компактность.

Следующий шаг — выбор микроконтроллера. Выбор был очевиден: Arduino Nano. Взял я его из-за небольших размеров, доступности и легкости в эксплуатации.

Перед проектировкой корпуса я решил выбрать компоненты и составить электронную схему. Чертеж выполнен в программе fritzing. В качестве «мозга» схемы я взял Arduino Nano. Высотометром стал барометр BMP180, который, находя разницу давлений в точке запуска и на максимальной высоте, по формуле считает расстояние и давление. Там стоит адаптер SD-карты и радиопередатчик, который передает послание на землю радиоволнами. В послании при помощи шифра Морзе скрыта высота. В следующих версиях будут установлены новые датчики.

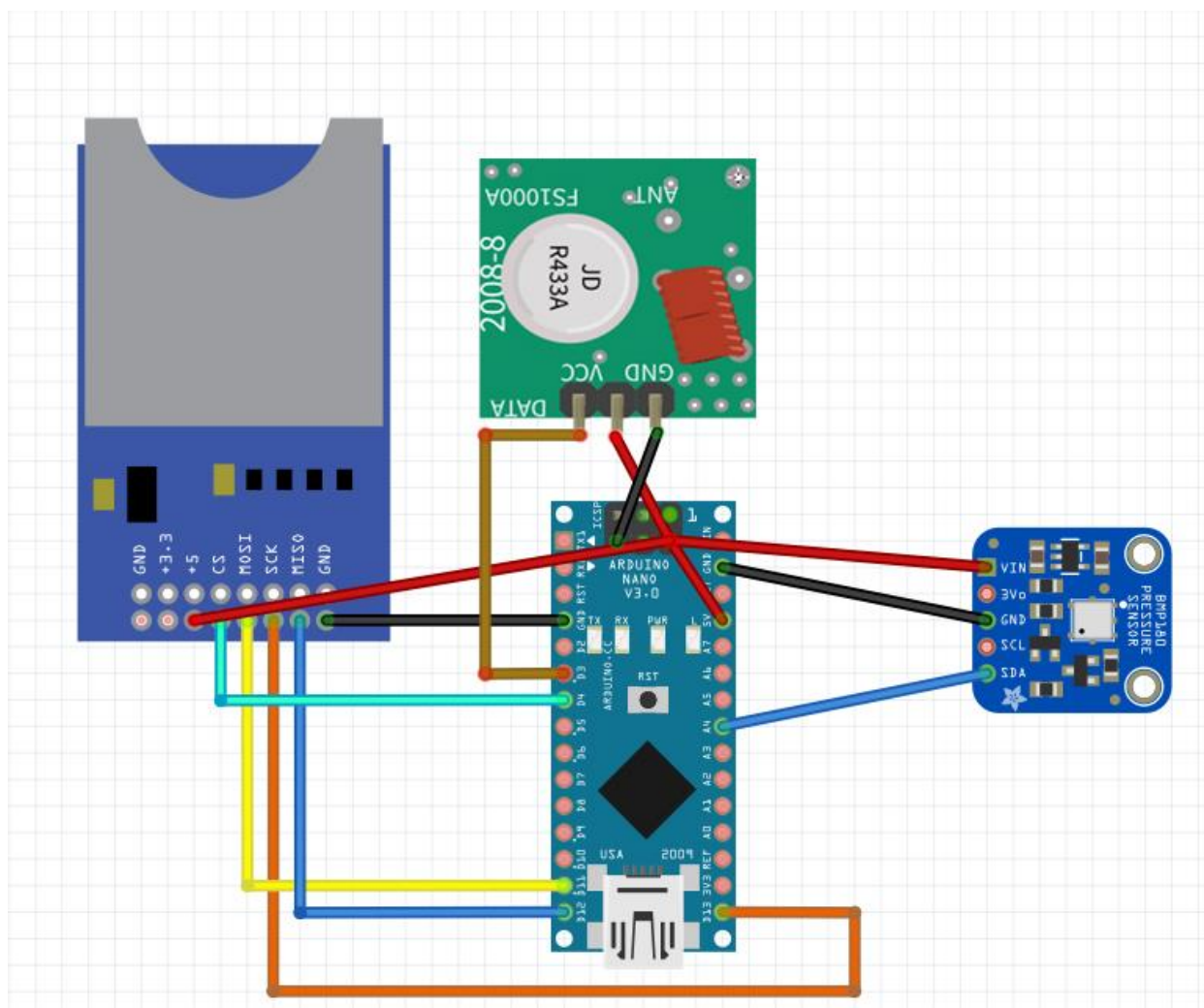


Рисунок 3. Схема электронной эмуляции

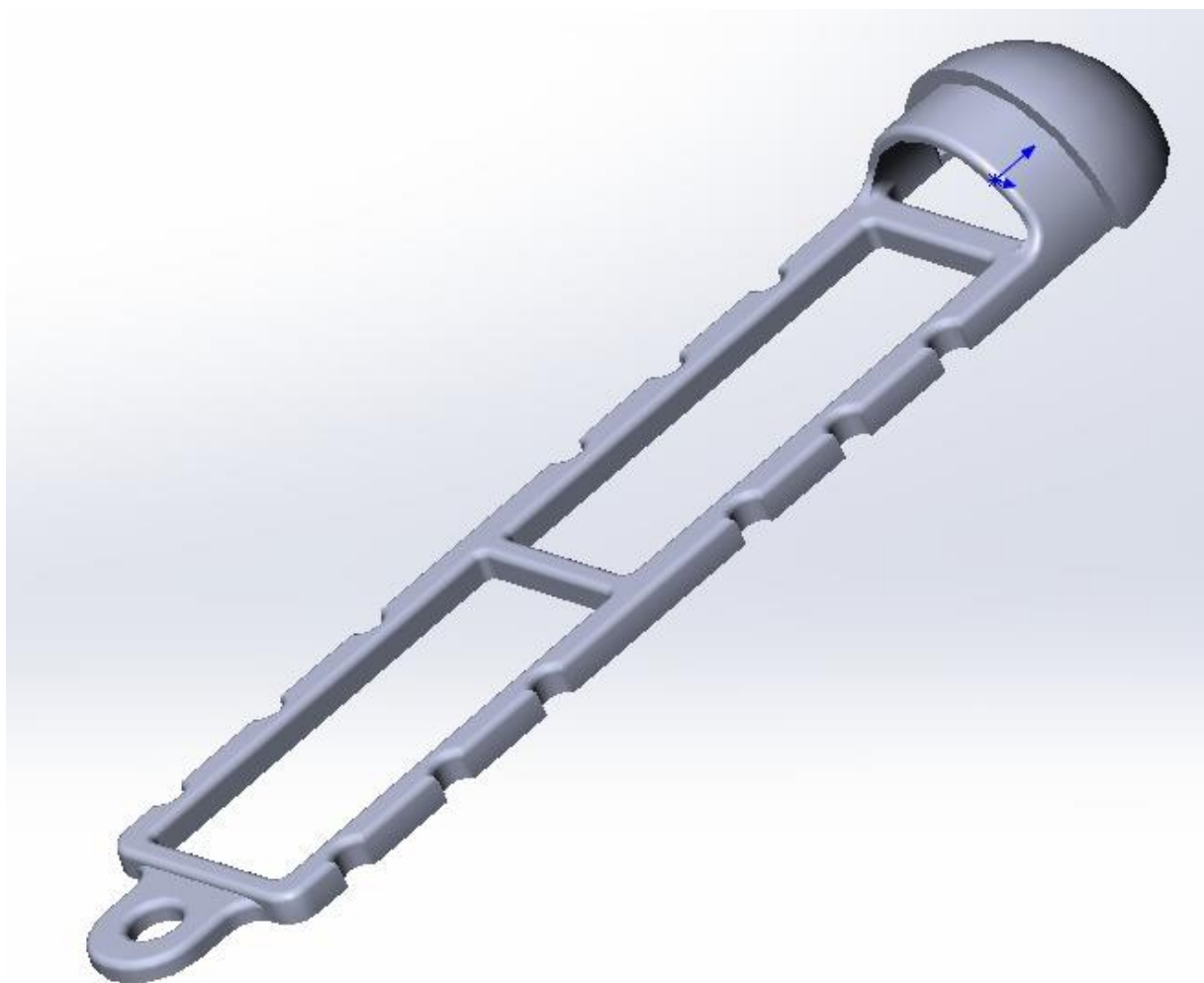


Рисунок 4. Трехмерная модель половины корпуса электроники вместе с обтекателем

6. Выбор технологии изготовления изделия. Технологическое описание процесса изготовления изделия.

При изготовлении моего изделия будут использованы разные материалы и информационные технологии на каждом из этапов: аддитивные технологии при изготовлении корпуса, пайка, программирование.

6.1. Технологическая карта.

6.1.1. Изготовление корпуса

Оборудование: ПК, 3D принтер Wanhao Duplicator 4S

Материалы: пластик PLA, ацетон, ветошь, наждачная бумага

Таблица 2. Изготовление корпуса

№	Содержание операции	Инструменты, приспособления
1.	Подготовка изображения и настройка экструдера. Выбор температурного режима под тип пластика, выбор плотности печати, работа в слайсере	Компьютер, программная оболочка Cura
2.	Печать корпуса	Компьютер, программная оболочка Cura, 3D принтер Wanhao Duplicator, пластик ABS
3.	Отделка изделия	Ацетон, ветошь, наждачная бумага

6.1.2. Сборка электронной части изделия

Оборудование: паяльная станция

Материалы: электронные компоненты, провода, припой, флюс

Таблица 3. Сборка электронной части изделия

№	Содержание операции	Инструменты, приспособления
1.	Припаять компоненты, согласно схеме	Паяльная станция, электронные компоненты, провода, припой, флюс

6.1.3. Программирование и пусконаладка
Оборудование: компьютер, программная оболочка ArduinoIDE, USB-кабель.

Таблица 4. Программирование и пусконаладка

№	Содержание операции	Инструменты, приспособления
1.	Написать программный код, согласно требованиям проекта	Компьютер, программная оболочка ArduinoIDE
2.	Загрузить скетч в контроллер и проверить работу программы	Компьютер, программная оболочка ArduinoIDE

6.2. Описание изготовления изделия

Изготовление изделия велось в соответствии с технологической картой. Сначала печать корпуса.

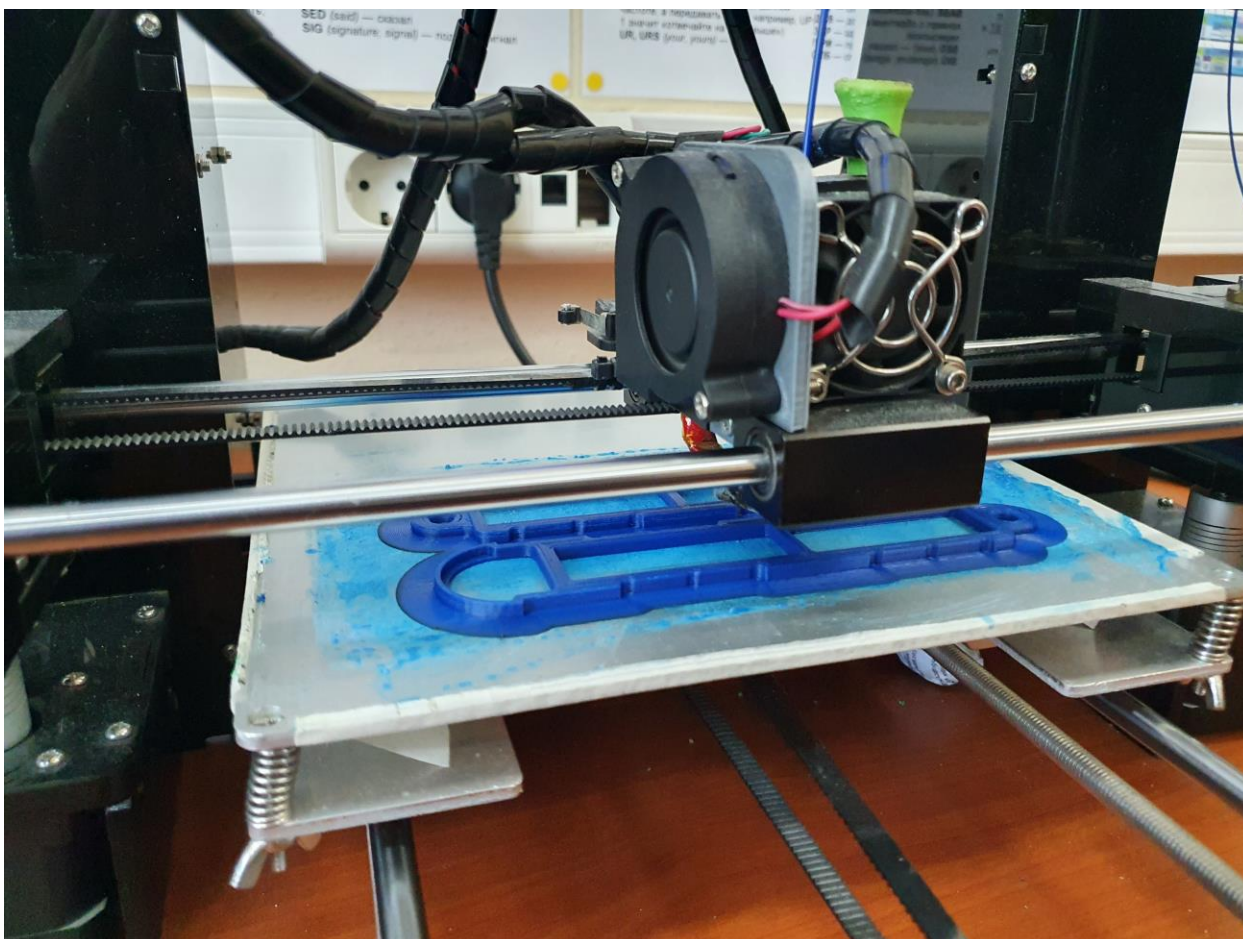


Рисунок 5. Печать корпуса

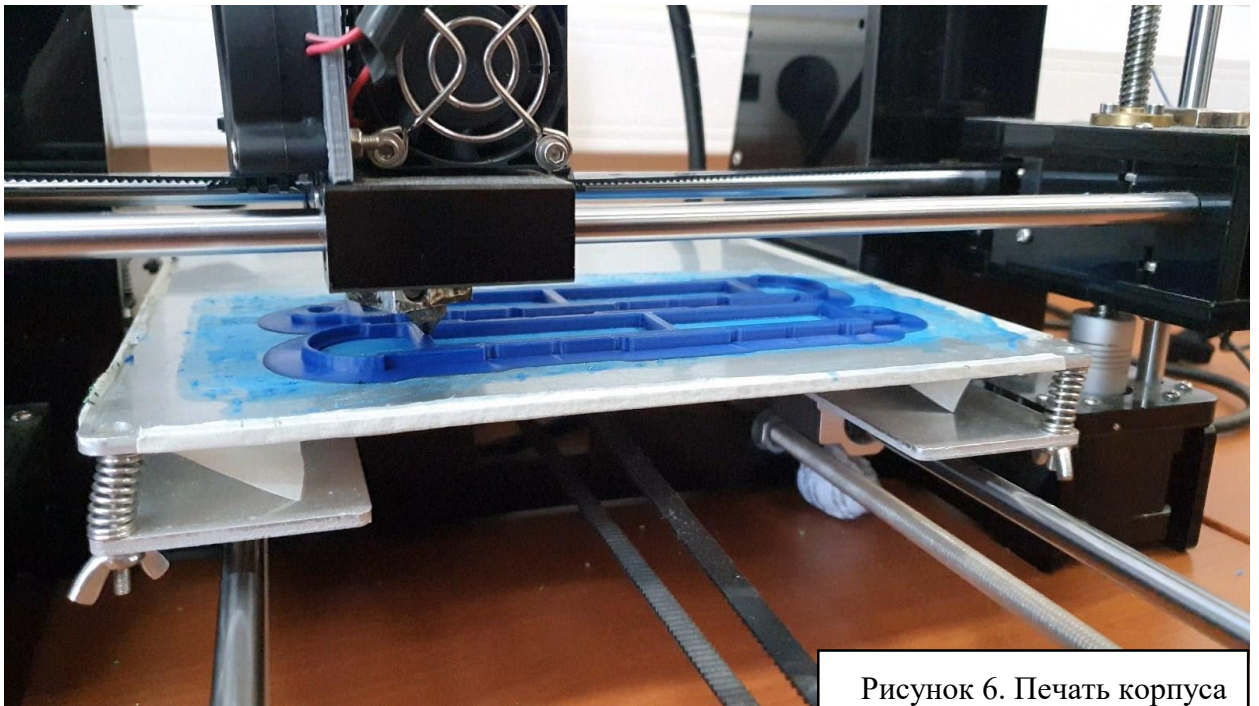


Рисунок 6. Печать корпуса

Следующий этап — пайка схемы. Пайка велась по технологической карте. И последним этапом стали программирование и пусконаладка. Во время пусконаладки электроника была помещена в корпус.

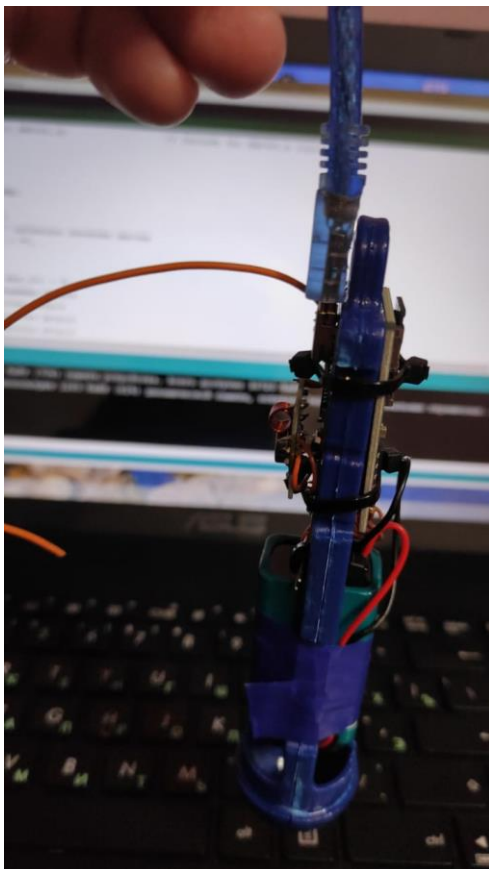


Рисунок 7. Пусконаладочные работы

7. Оригинальность предложенных технико-технологических, инженерных или эргономических решений, новизна проекта

Как итог была получена новая, улучшенная версия предыдущего проекта. Добавление функции измерения температуры уже расширило область применения ракеты, кроме как простых запусков.

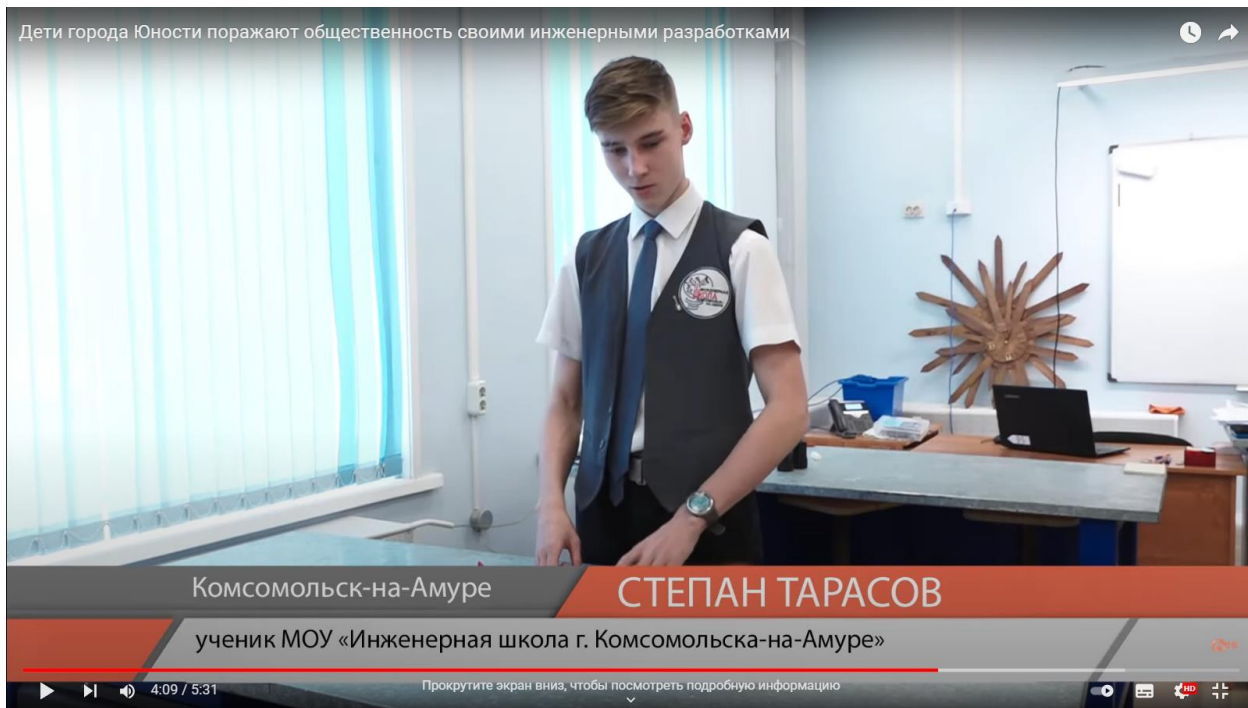

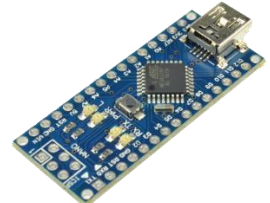
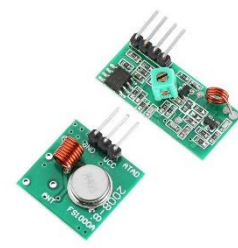
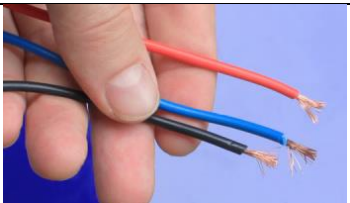






Рисунок 8. Запуск и интервью телеканалу 6ТВ

8. Экономическая и экологическая оценка будущего изделия и технология его изготовления

8.1. Расчет затрат на электронную составляющую проекта

Таблица 5. Расчет затрат на электронную составляющую проекта

№	Наименование	Фото	Цена	Кол-во	Стоимость, руб
1	Барометр BMP180		103.68	1	103.68
2	Микроконтроллер Arduino Nano		117.81	1	117.81
3	Беспроводной передатчик FS1000A		350	1	350
4	Провод RV1.5 – разных цветов		50	1	50
5	Батарейка Крона		200	1	200
6	Колодка для кроны		15	1	15
7	Адаптер для SD-карт		59	1	59

8	SD-карта 4 ГБ		103,27	1	103,27
Итого:					998,76

8.2. Расчет затрат на изготовление корпуса

Таблица 6. Расчет затрат на изготовление корпуса

№	Наименование	Цена, руб.	Кол-во	Стоимость, руб.
1	ABS-пластик	1600,00	14г	22,4
Итого:				22,4

8.3. Расчет энергозатрат при производстве

Таблица 7. Расчет энергозатрат при производстве

№	Наименование	P, Вт	Время использования, ч	Энергопотребление, кВт*ч	Стоимость, руб.
1	Ноутбук: черчение, программирование модели	48	2	0,38	0,76
2	3D принтер Wanhao Duplicator 4S	220	2,25	0,495	2,34
Итого:					3,1

*Стоимость электроэнергии *4р. 73к.* в соответствии с энерготарифом

**Косвенные затраты не рассчитывались (свет, отопление и т.д.)

8.4. Расчет трудозатрат при производстве

Таблица 8. Расчет трудозатрат при производстве

№	Наименование	Стоимость часа, руб.	Количество часов	Стоимость, руб.
1	Работа кодером, пайщиком	150	2	300
Итого:				300

*Учитывая, что я не высококвалифицированный специалист, то и стоимость часа моего труда будет оценена по МРОТ, т.е. 150 руб./ч

Внимание: это улучшенная версия предыдущей модели, поэтому время моей работы ушло лишь на обновление прошивки.

Полная стоимость изделия: 1324.3к.

Стоимость без трудозатрат: 1024р.3к.

8.5. Экологическая оценка

Мое изделие является максимально экологичным по следующим показателям:

1. Энергоэффективность (потребляет очень мало энергии);
2. Экологичность использования (не выделяет вредных веществ при эксплуатации);
3. Экологичность утилизации (большинство материалов могут быть подвержены вторичной переработке).

9. Рекламные предложения и перспективы внедрения изделия



Рисунок 9. Устройство в работе

Бортовая электроника заработала в ракете на конкурсе на космодроме «Восточный». Передатчик отправил сигнал Морзе, где была закодирована высота полета, но сама электроника отделилась от ракеты и была унесена на парашюте очень далеко. Именно для такого случая там стоял передатчик.



Рисунок 10. Призеры Всероссийских соревнований «Восточный старт»

Реклама в приложении Б.

10. Инструкция по использованию изделия:

1. Опустить устройство на минимальную высоту;
2. Замкнуть контакты батареи перемычкой;
3. Настроить рацию на нужную частоту;
4. Ждать момента, когда высота ракеты достигнет апогея;
5. Поймать сигнал Морзе для поздней дешифровки;
6. Если ракета приземлилась, вытащить флешку и вставить в компьютер.
7. Обработать полетные данные.

11. Список использованной литературы:

1. Сайт электроники <https://dvrobots.ru/>
2. Официальный сайт программы Fritzing
3. Сайт Росстат
4. Википедия. Статья про ракетомоделирование

Скетч

```

#include <Adafruit_BMP085.h>           // Include the BMP085.h library
#include <SPI.h>
#include <SD.h>

Adafruit_BMP085 bmp;

int SD_CS_PIN = 4;
float zero_alt; // начальное значение высоты
String dataString = "";
bool start, apog;
int i = 0;
int alt, alt_old, max_alt = 0;
int alt_m_x4; // тыщи метров
int alt_m_x3; // сотни метров
int alt_m_x2; // десятки метров
int alt_m_x1; // единицы метров

int temper10; // температура удесятерённая
int temper10_x3; // десятые доли градусов
int temper10_x2; // единицы градусов
int temper10_x1; // десятки градусов

int tone_pin = 3; // вывод для подключения передатчика, модулятор
int tone_freq = 800; // тональность для передачи апогея
int tone_freq_diag = 600; // тональность для самодиагностики
int dot_duration = 50;
int dash_duration;

unsigned long timeFly, timeStart;

int altitude()
{
  float temp_alt = 0;
  /*
  for (int i = 0; i < 3; i++) {
    temp_alt += bmp.readAltitude(); // Check if the measurement is complete
  }
  temp_alt = round(temp_alt/3);*/
  return round(bmp.readAltitude()); // Check if the measurement is complete
}

void saveSD() {
  // создаём файл для записи
  File dataFile = SD.open("datalog.txt", FILE_WRITE);
  // если файл доступен для записи
  if (dataFile) {
    // сохраняем данные
    dataFile.println(dataString);
    Serial.print(dataString);
  }
}

```

```
// закрываем файл
dataFile.close();
// выводим сообщение об удачной записи
Serial.println("Save OK");
} else {
// если файл не доступен
Serial.println("Error opening datalog.txt");
}
}
```

```
void cw_digit(int dig)
```

```
{
switch(dig)
{
case 1:
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
break;
case 2:
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
break;
case 3:
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
break;
case 4:
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
break;
case 5:
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
break;
case 6:
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
```

```

tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
break;
case 7:
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
break;
case 8:
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
break;
case 9:
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
break;
case 0:
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
break;
// default:
}
delay (dash_duration); delay (dash_duration);
}

```

void setup()

```

{
dash_duration = dot_duration*3;
pinMode(10, OUTPUT);
Serial.begin(9600);
if (!bmp.begin()) {
Serial.println("Could not find a valid BMP085 sensor, check wiring!");
tone_freq_diag = 200;
tone(tone_pin, tone_freq_diag);
while(1) {}
{}
}
// выводим сообщение в Serial-порт о поиске карты памяти
Serial.println("Initializing SD card...");
// если microSD-карта не была обнаружена

```

```

if (!SD.begin(SD_CS_PIN)) {
  // выводим сообщение об ошибке
  Serial.println("Card failed, or not present");
  tone_freq_diag = 1200;
  tone(tone_pin, tone_freq_diag);
  while(1) {}
  // don't do anything more:
  return;
} else {
  Serial.println("Card initialized");
}

```

```

zero_alt = altitude();
}

```

void loop()

```

{
temper10 = round(bmp.readTemperature()*10); // Check if the measurement is complete
alt = altitude() - zero_alt;
if (alt < 0) {alt = 0;}

if ((alt > 2) & !start)
{
  start = 1;
  timeStart = millis();
}
if (!apog) {tone(tone_pin, tone_freq_diag);}
if ((alt_old != alt) & start)
{
Serial.print("Alt = ");
Serial.println(alt);
dataString = (millis()-timeStart)*0.001;
dataString += "\t";
dataString += alt;
dataString += "\t";
dataString += temper10*0.1;
  // сохраняем данные с модулей на карту памяти
  saveSD(); // WRITE *** WRITE *** WRITE *** WRITE *** WRITE *** WRITE
*** WRITE *** WRITE ***
  // dataString = "";
  // delay(10);
if (alt > max_alt) { max_alt = alt; }

if ((alt > 2)&(alt < (max_alt-1)) & !apog)
{
  apog = 1;
  alt_m_x4 = max_alt*0.001;
  alt_m_x3 = max_alt % 1000 *0.01;
  alt_m_x2 = max_alt % 100 *0.1;
  alt_m_x1 = max_alt % 10;
}
}
}

```

```

if (apog)
{
// дудим букву М и передаём высоту апогея
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
//tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
delay (dash_duration); delay (dash_duration); delay (dash_duration);delay (dash_duration);
delay (dash_duration); delay (dash_duration);
if (alt_m_x4 != 0) cw_digit(alt_m_x4);
if ((alt_m_x4 != 0) or (alt_m_x3 != 0)) cw_digit(alt_m_x3);
if ((alt_m_x4 != 0) or (alt_m_x3 != 0) or (alt_m_x2 != 0)) cw_digit(alt_m_x2);
cw_digit(alt_m_x1);
delay (dash_duration); delay (dash_duration); delay (dash_duration);delay (dash_duration);
delay (dash_duration); delay (dash_duration);
// дудим букву А и передаём текущую высоту
tone(tone_pin, tone_freq); delay (dot_duration); noTone(tone_pin); delay(dot_duration);
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
delay (dash_duration); delay (dash_duration); delay (dash_duration);delay (dash_duration);
delay (dash_duration); delay (dash_duration);
alt_x4 = alt*0.001;
alt_x3 = alt % 1000 *0.01;
alt_x2 = alt % 100 *0.1;
alt_x1 = alt % 10;
if (alt_x4 != 0) cw_digit(alt_x4);
if ((alt_x4 != 0) or (alt_x3 != 0)) cw_digit(alt_x3);
if ((alt_x4 != 0) or (alt_x3 != 0) or (alt_x2 != 0)) cw_digit(alt_x2);
cw_digit(alt_x1);
delay (dash_duration); delay (dash_duration); delay (dash_duration);delay (dash_duration);
delay (dash_duration); delay (dash_duration);

// дудим букву Т и передаём текущую температуру
tone(tone_pin, tone_freq); delay (dash_duration); noTone(tone_pin); delay(dot_duration);
delay (dash_duration); delay (dash_duration); delay (dash_duration);delay (dash_duration);
delay (dash_duration); delay (dash_duration);
temper10_x3 = temper10 * 0.01;
temper10_x2 = temper10 % 100 * 0.1;
temper10_x1 = temper10 % 10;
cw_digit(temper10_x3);
cw_digit(temper10_x2);
cw_digit(temper10_x1);
delay (dash_duration); delay (dash_duration); delay (dash_duration);delay (dash_duration);
delay (dash_duration); delay (dash_duration);
}

alt_old = alt;

Serial.println(millis());
}

```