

## РАЗРАБОТКА И СОЗДАНИЕ ПО ДЛЯ МОДЕЛИ ЭЛЕКТРОМАГНИТНОЙ СТАБИЛИЗАЦИИ СПУТНИКА

Оразов А.В.

г. Орел, МБОУ СОШ № 50, 8 «А» класс

Руководитель: Демушкина О.В., г. Орел, МБОУ СОШ № 50,  
учитель информатики высшей квалификационной категории

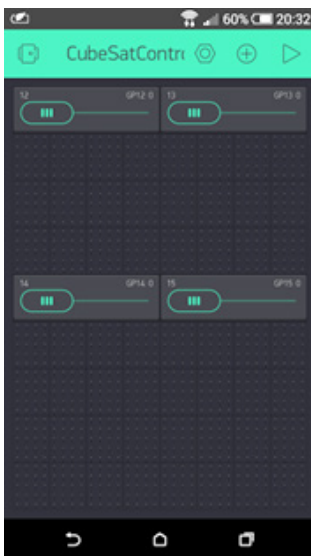
Реализация любого проекта зависит от его актуальности. В современном мире важна скорость разработки и внедрения продукта. Наша работа имеет право на существование, так как в ней реализована возможность быстрой модификации алгоритмов управления.

Цель: создать программное обеспечение (ПО) для управления моделью электромагнитной стабилизации спутника.

Этапы:

1. Изучение возможных способов управления моделью.
2. Создание ПО для прототипа модели.
3. Создание ПО для работы с законченным вариантом модели.
  - 3.1. Определение основных функций ПО.
  - 3.2. Выбор оптимальной среды для создания ПО.
  - 3.3. Создание ПО.
    - 3.3.1. Удаленное управление устройством. Коммуникационный интерфейс между устройством и браузером.
    - 3.3.2. Развертывание файловой системы на модуле.
    - 3.3.3. Определение положения устройства и регулировка его позиции в трехмерном пространстве.
4. Вывод.

### 1. Изучение возможных способов управления моделью



Перед тем как начать создавать ПО, мы изучили возможные способы управления моделью. Для прототипа спутника мы выбрали Wi-Fi модуль ESP8266, который может быть запрограммирован и использован для контролирования модели. Для быстрого старта и изучения поведения модели, мы воспользовались программой “Blynk”, с помощью которой можно было управлять выходами Wi-Fi модуля напрямую с телефона.

### 2. Создание ПО для прототипа модели

Настроить управление моделью при помощи “Blynk” оказалось очень просто, удобный и понятный интерфейс сделали этот этап не сложным.

Во время сборки модели (приложение 1, рис. 1) мы нашли один минус текущего варианта Wi-Fi модуля – у него было недостаточно выводов (GPIO), поэтому нам пришлось проводить эксперименты с прототипом, используя только две пары катушек. Впрочем, для выявления недостатков модели этого было достаточно. Схема прототипа представлена на рисунке 2 в приложении 1.

Разобравшись с проблемами прототипа, мы приступили к созданию окончательного варианта модели.

### 3. Создание ПО

#### 3.1. Определение основных функций ПО



В конечном варианте модели мы использовали Wi-Fi модуль ESP – WROOM – 32, который обладал достаточным количеством выводов, чтобы управлять сразу тремя парами катушек. Его вычислительной мощности хватило, чтобы обрабатывать запросы веб-сервера, работу с внутренней файловой системой и производить обработку внешних входов-выходов для считывания данных гироскопа и управления катушками. Далее

были обозначены требования к функциям нашего ПО:

1. Использование платформы Arduino для написания программы.
2. Веб-сервер. Возможность загрузки html страниц удаленно.
3. Раздельное ШИМ управление 6 электромагнитами.
4. Пропорциональное регулирование парами катушек в зависимости от данных с гироскопа.

### 3.2. Выбор оптимальной среды для создания ПО



ПО мы создавали в среде программирования VisualStudio 2015 (VS 2015) со встроенным плагином VisualMicro, который позволил работать нам с нашим модулем как с Arduino. Создание веб-страниц и их отладка производились в отдельном веб-приложении. Готовые страницы загружались удаленно на устройство. Использовались языки программирования C++ для написания программы модуля ESP32, HTML и JavaScript для создания клиентской части веб-сайта.

### 3.3. Создание ПО

#### 3.3.1. Удаленное управление устройством. Коммуникационный интерфейс между устройством и браузером

Wi-Fi модуль был сконфигурирован как точка доступа. В программной части веб-сервера были созданы методы для обработки запросов: загрузка/удаление файлов, статус файловой системы, данные с гироскопа, выставление ШИМ значений на катушках. Для запросов, которые не определены в программе, делается попытка спроецировать их на список файлов, и если такой файл существует, то он выдается как результат запроса.

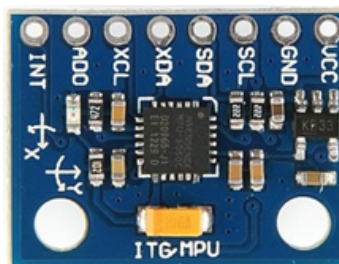
#### 3.3.2. Развертывание файловой системы на модуле

Для создания хранилища файлов на самом Wi-Fi модуле без применения дополнительного оборудования была применена библиотека SPIFFS, которая использовала в качестве хранилища часть объема

микросхемы памяти. Управление файлами осуществляется через веб-страницу (приложение 2, рис. 1). Доступны методы для файлов: удалить, добавить, открыть и отформатировать все файловое хранилище.

С помощью этой системы мы впоследствии загрузили на плату веб-страницу (приложение 2, рис. 2) с отображением модуля в пространстве, куб, который вращался в соответствии с положением реального модуля.

#### 3.3.3. Определение положения устройства и регулировка его позиции в трехмерном пространстве



Чтобы осуществить пропорциональное регулирование модели, в качестве датчика положения в пространстве мы использовали гироскоп MPU6050. С его помощью мы практически полностью удалили раскачивание в разные стороны у модели при разворотах.

Так же нами был выявлен минус использования подобного гироскопа в единственном экземпляре. Оказалось, что его диапазон значений ограничен 180 градусами по всем трем плоскостям, и при развороте более 180 градусов, он возвращал некорректные значения.

Для опытного образца мы обозначили 4 позиции. По умолчанию модель смотрит лицевой стороной на нас. Каждая следующая позиция – это поворот по оси X на 90/180 градусов влево или вправо. Такой вариант оказался самым оптимальным при использовании всего лишь одного гироскопа. Законченный вариант модели и его схема изображены в приложении 3, рисунок 1 и 2.

#### Приложение 1



Рис. 1. Вид прототипа модели

CubeSAT settings.

Список файлов на устройстве.

ИМЯ	РАЗМЕР	Удалить	Открыть
/rotact.html	2990 B	<a href="#">Удалить</a>	<a href="#">Открыть</a>
/jquery-1.10.2.min.js	94140 B	<a href="#">Удалить</a>	<a href="#">Открыть</a>
/cube.html	9187 B	<a href="#">Удалить</a>	<a href="#">Открыть</a>
/pid_coeff.data	24 B	<a href="#">Удалить</a>	<a href="#">Открыть</a>
/reg_log.csv	14825 B	<a href="#">Удалить</a>	<a href="#">Открыть</a>
/coils.html	3305 B	<a href="#">Удалить</a>	<a href="#">Открыть</a>

Загрузка файлов на устройство.

Всего: 1374476 B; Свободно: 1246717 B

Выберите файл:  | No file chosen

[Отформатировать](#)

W: ? X: ? Y: ? Z: ? Is OK: Error

Состояние устройства.

Тери, С: 40.56 Тгуно, С: 34.34 Состояние: good

Рис. 1. Веб-страница настроек

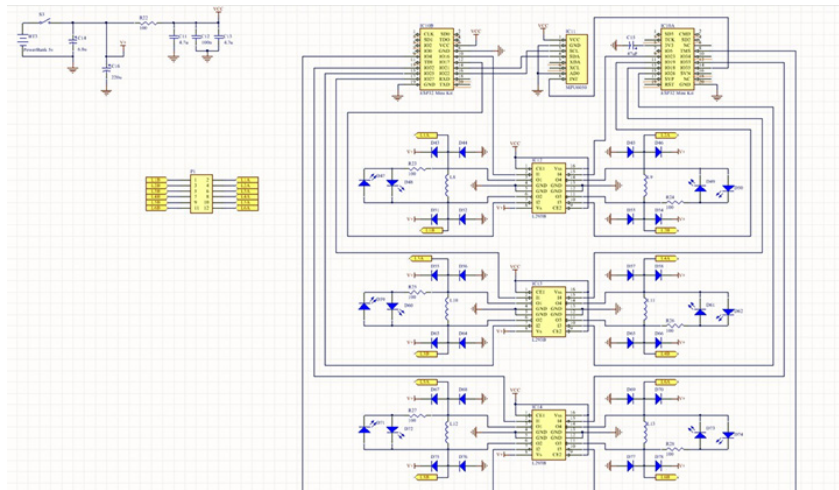


Рис. 1. Схема платы управления готового образца

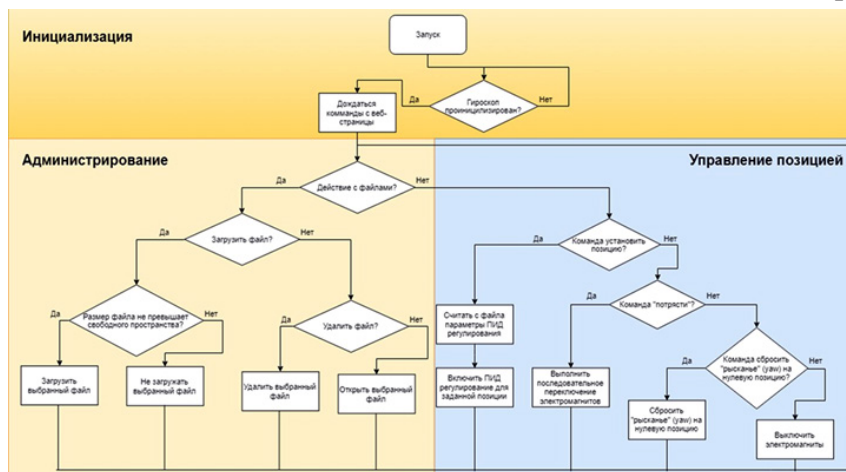


Рис. 1. Алгоритм работы ПО

### 3.4. Вывод

Результатом нашей работы стало ПО, которое позволяет изменять алгоритмы позиционирования модели без необходимости перепрограммирования самого устройства. Всю логику можно прописать в пользовательской веб-странице на JavaScript. Алгоритм работы ПО изображен в приложении 4, рисунок 1.

### Список литературы

1. Espressif Systems – esp32\_hardware\_design\_guidelines\_en.pdf.
2. Espressif Systems – esp-wroom-32\_datasheet\_en.pdf.
3. InvenSense Inc. – MPU-6000-Datasheet1.pdf.
4. InvenSense Inc. – MPU-6000-Register-Map1.pdf.
5. Neil Kolban – Kolban’s Book on the ESP32 & ESP8266.
6. Интернет-ресурс Geektimes – статья “ESP32 и Ардуино” <https://geektimes.ru/post/290205/>