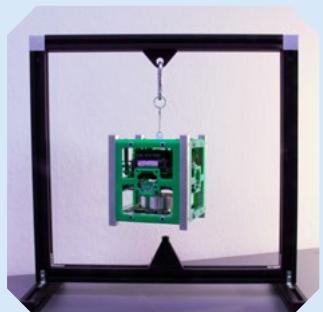
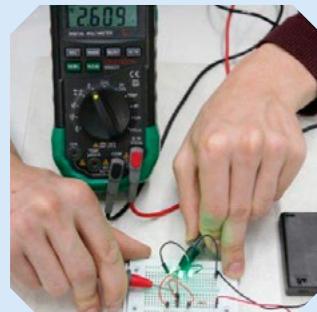
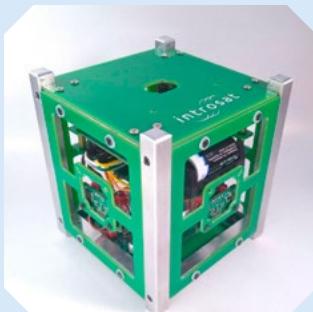
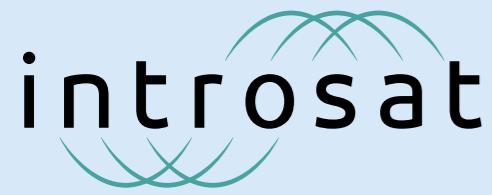


Спутникостроение

Учебное пособие. Под редакцией К. Ю. Якушиной, Е. А. Вениной, ноябрь 2020 г.





ДАТЧИКИ НА СПУТНИКЕ
ИСПОЛНИТЕЛЬНЫЕ УСТРОЙСТВА
ОРИЕНТАЦИЯ КОСМИЧЕСКИХ АППАРАТОВ

ДАННОЕ ПОСОБИЕ РАССМАТРИВАЕТ ОБЩЕЕ УСТРОЙСТВО КОСМИЧЕСКИХ АППАРАТОВ, А ТАКЖЕ ЯВЛЯЕТСЯ РУКОВОДСТВОМ ДЛЯ ПОСТРОЕНИЯ КУРСА ПО СПУТНИКОСТРОЕНИЮ НА БАЗЕ КОНСТРУКТОРА INTROSAT.

В ХОДЕ ЗАНЯТИЙ УЧАЩИЕСЯ СМОГУТ ИЗУЧИТЬ ПРИНЦИПЫ РАБОТЫ ИЗМЕРИТЕЛЬНЫХ УСТРОЙСТВ И СПОСОБЫ ОБРАБОТКИ ПОСТУПАЮЩЕЙ ОТ НИХ ИНФОРМАЦИИ, ОСВОИТЬ ОСНОВЫ ПРОГРАММИРОВАНИЯ МИКРОКОНТРОЛЛЕРА И ПОНЯТЬ ЛОГИКУ ВЫСТРАИВАНИЯ УПРАВЛЯЮЩИХ КОМАНД, ПОЗНАКОМИТЬСЯ С ЗАДАЧЕЙ СТАБИЛИЗАЦИИ КОСМИЧЕСКОГО АППАРАТА В УСЛОВИЯХ НЕВЕСОМОСТИ И ЕГО ПОСЛЕДУЮЩЕЙ ОРИЕНТАЦИИ В ПРОСТРАНСТВЕ.

КОНСТРУКТОР INTROSAT ПОЗВОЛЯЕТ ИЗУЧИТЬ АСПЕКТЫ КОНСТРУИРОВАНИЯ КОСМИЧЕСКИХ АППАРАТОВ И ОСУЩЕСТВИТЬ СБОРКУ ФУНКЦИОНАЛЬНОГО МАКЕТА СПУТНИКА ФОРМАТА CUBESAT. ДОПОЛНЕНИЯ К НАБОРУ ПОЗВОЛЯЮТ РАСШИРЯТЬ СПИСОК ИЗУЧАЕМЫХ ПОДСИСТЕМ И ПРИНЦИПОВ РАБОТЫ СПУТНИКОВ.

Рекомендации

Рекомендуемый размер группы:

от 3 до 15 учащихся, по 3-4 человека на набор компонент.

Уровень подготовки:

с 8-го по 11-й классы, возможна

адаптация курса для 7-го класса и младше.

Базовые знания, которые помогут лучше усвоить курс:

основные понятия в электричестве и электромагнетизме; законы вращательного движения, угловая скорость; инерция, закон сохранения энергии; первая космическая скорость, закон всемирного тяготения; основы алгоритмизации и программирования (переменные, циклы, условные операторы).



Школьный курс космонавтики
Серия основана в 2017 году

К. Ю. Якушина, Н. М. Яковлева, Н. А. Муллин, С. А. Кусков

Спутникостроение

Учебное пособие

Под редакцией К. Ю. Якушиной, Е. А. Вениной

Москва
«Образование Будущего»
2021

УДК 629.783
ББК 39.64
Я499

СОДЕРЖАНИЕ

Предисловие от авторов	4
Техника безопасности при работе с конструктором IntroSat	5
Требования безопасности	5
Рекомендации при работе с конструктором IntroSat	5
Часть I: Введение	6
Космические аппараты: что это такое?	6
Часть II: Небесная механика	12
Движение по орбите	13
Расчет кеплеровых элементов орбиты космического аппарата по заданным условиям	23
Часть III: Датчики на спутнике	27
Разбираемся в принципе работы потенциометра	28
Принцип работы датчика освещенности	37
Сборка стенда солнечной батареи	37
Программирование стенда умной солнечной батареи	42
Часть III: Исполнительные устройства	47
Способы ориентации космических аппаратов	48
Исследование стабилизации с использованием двигателя-маховика	54
Алгоритм стабилизации	61
Часть IV: Ориентация в космическом пространстве	64
Способы ориентации КА	64
Изучение ориентации по свету	67
Приложение 1. Спецификация набора	73
Приложение 2. Инструкция по сборке подвеса	76
Приложение 3. Прошивка микроконтроллера через bluetooth-модуль	78
Список используемых источников	81

ПРЕДИСЛОВИЕ ОТ АВТОРОВ

4

Данное пособие предназначено для ведения занятий по космонавтике с теоретическими обзорами и практикумами на основе учебных конструкторов спутников семейства IntroSat™. Пособие входит в состав серии «Космическая профориентация», в совокупности составляющей комплексный курс инженерии прикладных космических систем для школьников и студентов; рассматриваемые в пособии учебные модули рекомендуются как начало соответствующего курса или как самостоятельный вводный курс спутникостроения. Также, рассматриваемые темы и практические навыки входят в рекомендованную программу подготовки профиля «Спутниковые системы» Национальной Технологической Олимпиады и конкурсов серии «Прикладные космические системы» программы «Дежурный по планете».

Пособие ориентировано для преподавателей, но может быть использовано и учащимися.

В практической части учащимся предстоит работать с микроконтроллерами семейства STM32, широко распространенными в промышленности, в том числе в космической отрасли. Задания в составе курса развивают навыки программирования на примере прикладных задач и в практическом применении к современному оборудованию.

Применяемое в составе конструктора оборудование позволяет работать как в средах разработки, специально созданных для семейства STM32, так и в схожей, но более простой учебной среде Arduino IDE, что позволяет при необходимости адаптировать уровень задач. В качестве материалов, дополняющих данное пособие, можно использовать широко распространенные в интернете руководства и примеры по работе с Arduino.

Так как работа с оборудованием требует некоторой сноровки, то мы настоятельно рекомендуем перед тем, как начинать проводить занятия, пройти все этапы работы с ним самостоятельно.

В издании 2021 года учтен опыт реализации практикумов в составе курса за период с 2019 по 2021 год, а также последние версии компонентов конструкторов IntroSat™, актуальные на 2 полугодие 2021 года.

Загружаемые материалы по темам курса (презентации, библиотеки и прошивки для оборудования) доступны на сайте introsat.ru.

Настоящий курс активно развивается, и нам важно Ваше мнение и обратная связь. Если что-то не получается и/или Вы хотите больше узнать об актуальных возможностях для интересующихся космонавтикой школьников и студентов (образовательные программы, соревнования, проектные смены и другое), то пишите нам вопросы и обращения на почту info@introsat.ru, мы обязательно поможем.

ТЕХНИКА БЕЗОПАСНОСТИ ПРИ РАБОТЕ С КОНСТРУКТОРОМ INTROSAT

5

ТРЕБОВАНИЯ БЕЗОПАСНОСТИ

Входящее в состав конструктора оборудование ориентировано на обучение прикладным инженерным навыкам (включая навыки разработки электрических схем и пайки), имеет открытую архитектуру и допускает совместную работу со многими сторонними компонентами. При этом работоспособность электронных компонентов и безопасность труда может быть обеспечена только при условии соблюдения соответствующих требований:

1. Не допускается работать с розетками, в которых заземление (или расширитель системы защиты от поражения электрическим током) отсутствует, или не соответствует установленным стандартам.
2. Не допускается подключение устройств к напряжению, не соответствующему указанному номиналу.
3. Не допускается подключение устройств к оборудованию (в т.ч. ПК), подключенному к незаземленной сети.
4. Не допускается использование печатных плат при их контакте с водой или иной токопроводящей жидкостью, в т.ч. с ее остаточными следами на поверхности.
5. Не допускается работа с оборудованием в помещениях с высокой влажностью.
6. Не допускается расположение плат на металлической или металлизированной поверхности при наличии на них напряжения.
7. Не допускается касание оголенных участков проводов незащищенными руками.
8. Не допускается отключение/подключение периферийных устройств к цепи с микроконтроллером при наличии в ней напряжения.
9. Не допускается использование сторонних источников питания.
10. При выявлении неисправности необходимо сразу же отключить устройство от питания.

11. При построении электрических цепей необходимо соблюдать правила построения электрических цепей и избегать появления короткого замыкания.

12. При коммутации плат по нескольким контактам необходимо внимательно следить за правильностью соединения контактов.

13. Не допускается эксплуатация аккумуляторов при наличии у них признаков механического повреждения, протекания и/или вздутия.

14. Не допускается эксплуатация аккумуляторов за пределами рабочего диапазона температур.

РЕКОМЕНДАЦИИ ПРИ РАБОТЕ С КОНСТРУКТОРОМ INTROSAT

1. Рекомендуется использование приемов защиты от статического электричества в процессе работы с электронными устройствами в составе набора (использование антистатических поверхностей, инструментов, браслетов).

2. Перед прошивкой отладочной платы Blue Pill через ST-Link рекомендуется отключать плату управления маховиком и модуль питания от материнской платы.

3. Заряжать аккумулятор рекомендуется помощью зарядного устройства, идущего в комплекте, или в составе модуля питания. При зарядке загорается желтый диод. Когда аккумулятор заряжен, загорается зеленый диод.

Будьте осторожны при подключении сторонних устройств. Конструктор спроектирован таким образом, что подключение ардуино-совместимой периферии к платам допустимо. Однако обратите внимание на крепления сторонних устройств – они не должны касаться компонентов плат. Для того, чтобы это предотвратить, можно воспользоваться изолирующими шайбами, которые идут в наборе.

ЧАСТЬ I

Теоретическая часть



2 ЧАСА

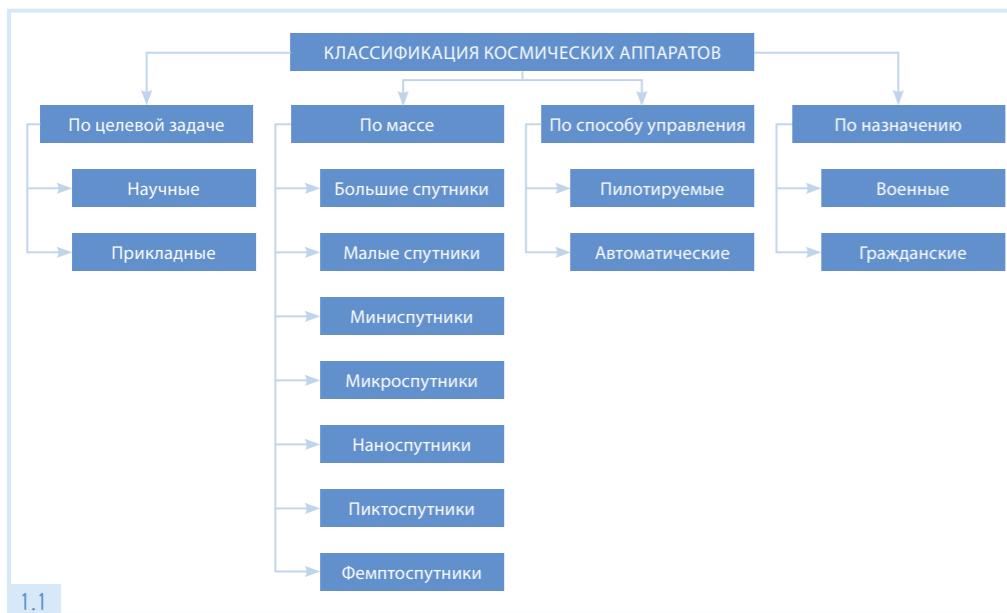
Космические аппараты: что это такое?

СОВЕТЫ УЧИТЕЛЮ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ. В первой части занятия необходимо сделать акцент на разнообразии задач, решаемых космическими аппаратами. Помимо искусственных спутников Земли можно рассказать о других видах космической техники: ракетах-носителях, планетоходах, космических кораблях и т. д. Не лишним будет отметить, что спутник является частью инфраструктуры, в которую входят также стартовые комплексы и наземные станции приема данных. Важно рассказать о назначении космических аппаратов и показать, как оно определяет их строение и внешний вид. Например, если речь идет об аппаратах дистанционного зондирования Земли (ДЗЗ), то они, как правило, имеют вытянутую форму из-за удлиненной конструкции оптической аппаратуры; а спутники связи тем больше, чем дальше расположены от поверхности Земли, и в их строении часто можно распознать антенны. При проведении занятия вы также можете воспользоваться материалами презентации «Вводная лекция», расположенной по ссылке: introsat.ru/download.

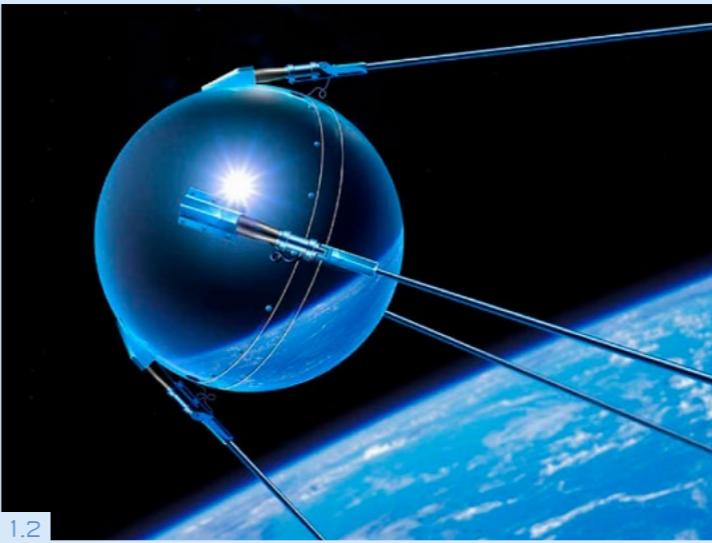
ВВЕДЕНИЕ И ТЕОРИЯ

На сегодняшний момент искусственные спутники получили широкое распространение благодаря ряду своих особенностей, а именно: быстрый проход над поверхностью Земли, движение по орбите за счет

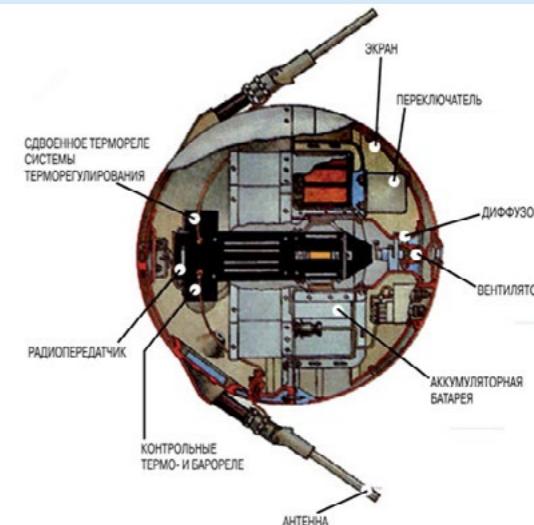
инерции, не требующее затрат топлива после выведения, нахождение вне атмосферы. Все это способствовало распространению применения космических аппаратов в ряде научных и прикладных задач.



1.1 КЛАССИФИКАЦИЯ КОСМИЧЕСКИХ АППАРАТОВ



1.2



Помимо получения снимков поверхности, спутники решают задачи связи, интернета и телевидения, играя роль ретранслятора: передают сигнал из одной точки Земли в другую. Несколько таких спутников могут обеспечить покрытие всего земного шара.

Первый искусственный спутник был запущен в 1957 году и представлял из себя, по нынешним меркам, простое устройство, назначением которого была всего одна задача – передавать простой сигнал в пространство.

Значительно сложнее принцип, по которому работают современные навигационные группировки: по полученным с нескольких спутников сигналам определяется местоположение объекта, который эти сигналы принимает. На данный момент существует две глобальные навигационные системы: GPS (США, 32 действующих аппарата в группировке) и ГЛОНАСС (Россия, 24 действующих аппарата в группировке). Еще две активно разворачиваются на данный момент: Galileo (ЕС) и Baidou (Китай).

Однако навигация – не единственное применение космических аппаратов. Также существуют спутники дистанционного зондирования Земли (ДЗЗ), на борту которых стоят камеры или иные устройства, позволяющие снимать поверхность нашей планеты. Благодаря получаемым с таких аппаратов снимкам, можно следить за состоянием посевов, движением льдов, выявлять очаги возгорания, составлять прогнозы погоды, обновлять карты местности и многое другое.

ЗАДАЧИ КОСМИЧЕСКИХ АППАРАТОВ

Прикладные задачи	Научные задачи
<ul style="list-style-type: none"> • навигация • картографирование • мониторинг поверхности • связь 	<ul style="list-style-type: none"> • астрономические наблюдения • космическая погода • изучение влияния невесомости, вакуума, радиации • межпланетные миссии • технологические исследования и эксперименты

1.2 ПЕРВЫЙ
ИСКУССТВЕННЫЙ
СПУТНИК ЗЕМЛИ

СОВЕТ. Предложите сперва учащимся самим назвать виды и примеры задач для космических аппаратов.

Множество прикладных задач для спутников привело к их широкому распространению, несмотря на высокую стоимость запусков. На данный момент на орбите Земли находится несколько тысяч космических аппаратов.

Помимо спутников, работающих на околоземных орбитах, запускают и автоматические станции для изучения планет Солнечной системы и за ее пределы. Аппарат «Кассини» долгое время собирал информацию о Сатурне, «Юнона» изучает Юпитер, а «Новые горизонты» впервые сделал детальный снимок Плутона. Существуют также аппараты, запущенные за пределы Солнечной системы, такие как «Вояджер» 1 и 2.

ВЫВОД СПУТНИКА НА ОРБИТУ

Для того чтобы спутник вывести на орбиту, ему необходимо сообщить первую космическую скорость.

Первая космическая скорость – это та скорость, которая позволяет объекту выйти на замкнутую траекторию движения вокруг центра Земли. Для нашей планеты первая космическая скорость равняется 7,9 км/с, что несложно вычислить из простого соотношения.

За пределы атмосферы спутники выводятся с помощью ракет-носителей, способных придать им космическую скорость за счет мощных двигателей. Во время выведения на орбиту космический аппарат подвергается ряду физических нагрузок: перегрузке, вибрациям и ударам.

Перегрузка – явление, при котором возрастает вес. В обычных условиях на поверхности Земли действующая на тела перегрузка равна ускорению свободного падения ($1g$), что определяется притяжением к Земле. Однако при ускоренном движении (разгоне, торможении или повороте) перегрузка начинает расти. Из-за огромной силы тяги двигателей перегрузка на ракетах может достигать $20g$. То есть, спутник, который на Земле весил 10 кг, в таком полете будет весить 200 кг. Понятно, что корпус космического аппарата и его бортовую аппаратуру необходимо делать достаточно прочными для того, чтобы значительное увеличение веса не привело к его поломке.

СОВЕТ. Спросите или напомните учащимся, чем отличаются вес и масса. Обсудите, одинаково ли ускорение свободного падения на разных планетах.

Вибрации в полете создает ракетный двигатель, в котором в одну секунду сгорают тонны окислителя и горючего. Горение практически неизбежно сопровождается пульсациями давления, которые преобразуются в пульсации тяги. Изменения тяги приводят к циклическому изменению нагрузки на спутник, в результате чего его конструкция начинает вибрировать. Воздействие вибраций разрушительно само по себе, однако главная опасность – входжение спутника в резонанс. Резонанс – явление, при котором частота внешних вибраций совпадает с внутренними колебаниями тела. Известны случаи, когда под действием резонанса обрушивались мосты. Поэтому конструкция аппарата должна быть устойчива к воздействию вибраций и резонанса.

Удары при полете на ракете возникают при отделении от нее различных частей – отработавших ступеней и спутников. Часто подобные операции совершаются с помощью пиротехнических устройств, внутри которых происходят небольшие, тщательно рассчитанные взрывы, как раз и вызывающие удары. Удары, как и вибрации, и перегрузки – это увеличение сил, действующих на спутник, а их чрезмерная величина может привести к поломке.

НАХОЖДЕНИЕ В КОСМИЧЕСКОМ ПРОСТРАНСТВЕ

Уже находясь в космическом пространстве, космический аппарат также подвергается воздействиям среды.

Во-первых, космический аппарат находится в невесомости. Невесомость, с одной стороны, помогает работе спутника, так как на все его части перестают действовать силы веса, и любая часть космического аппарата может совершать движение под действием самой малой силы. С другой стороны, в невесомости жидкое ракетное топливо всплывает и собирается внутри баков в огромные капли, свободно движущиеся по всему объему. Если в этот момент попытаться включить двигатель, то в него пойдет газ и никакой полезной работы он совершать не сможет. Кроме того, так как в невесомости силы веса не действуют, то нет и силы Архимеда; а значит, теплый воздух остается там же, где и холодный, и они не перемешиваются, что тоже создает проблемы. Помимо этого, невесомость приводит к негативным изменениям в живых организмах. Без физических нагрузок и сил веса атрофируются мышцы, из костей вымывается кальций и развитие эмбрио-

нов животных происходит с серьезными нарушениями.

Невесомость возникает в том случае, когда тело движется только под действием силы тяжести и никакие иные силы к нему не приложены. На орбите спутник движется, в основном, под действием силы притяжения; влияние других сил чрезвычайно мало. На Земле это явление встречается в свободном падении.

Во-вторых, космический аппарат находится вне атмосферы, то есть в условиях вакуума. В вакууме с веществами происходит множество необычных явлений. Многие жидкости начинают интенсивно испаряться (испаряются даже твердые тела вроде красок, пластмасс и резин). Более того, даже металлы медленно переходят в газообразную форму. При очень низком давлении возможно появление вакуумной сварки, когда части механизмов спутника за небольшое время очень плотно слипаются, молекулы проникают из одной детали в другую, и бывшее подвижным соединение застывает. Немаловажно также, что, так как вакуум – это отсутствие среды, в нем невозможна передача тепла никаким иным способом, кроме как посредством излучения.

Вакуумом называют область пространства с давлением, пониженным относительно некоего нормального уровня: например, ниже атмосферного. Чем ниже давление, тем глубже вакуум. При совсем глубоком вакууме говорят уже не о давлении как таковом, а о количестве частиц в единице объема.

В-третьих, космический аппарат подвержен сильному перепаду температур. На орбите Солнце светит в четыре раза ярче, чем в самый ясный день на Земле. Это приводит к тому, что освещенные части космического аппарата могут нагреться до очень большой температуры, а затененные – очень сильно остывать. При значительном и быстром перепаде температур (так называемом «тепловом ударе») многие материалы расширяются, сжимаются, растрескиваются. Но даже медленное изменение температуры за рабочие диапазоны самих механизмов спутника или его бортовой аппаратуры способно привести к их поломке.

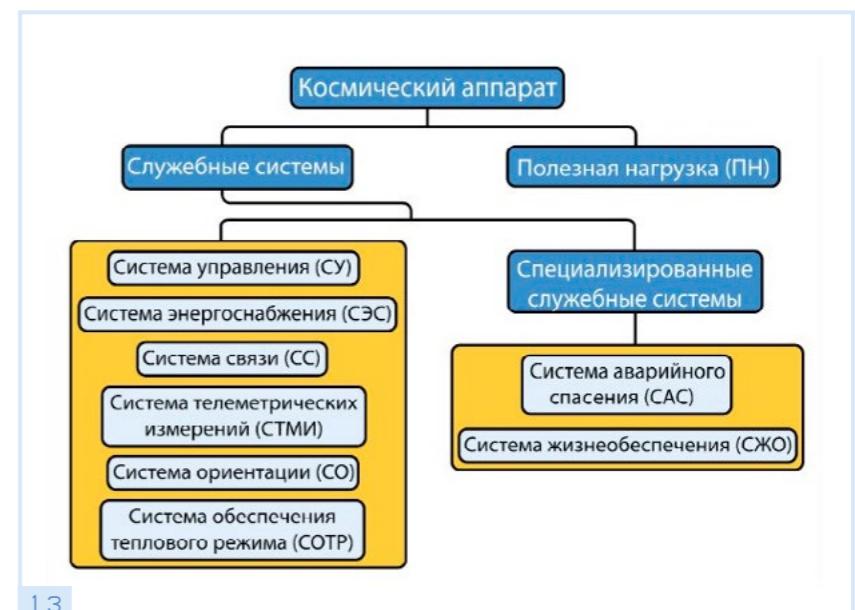
В-четвертых, аппарат подвержен воздействию радиации. Кроме видимого света еще есть излучения как с меньшей длиной волн (ультрафиолетовое, рентгеновское и гамма-излучение), так и с большей

(инфракрасное и радио). Коротковолновое излучение переносит немного энергии и обычно не сильно нагревает спутник, но зато оно может приводить к ухудшению свойств материалов спутника и выходу из строя электроники. Длинноволновое излучение может создавать помехи работе системы радиосвязи космического аппарата.

Все это предполагает ряд серьезных требований к конструкции аппарата.

СОСТАВ СПУТНИКА

Вне зависимости от своего назначения, устройства спутника образуют сложную техническую систему, которая, в свою очередь, состоит из подсистем. Подсистемы космического аппарата, как правило, подразделяются на полезную нагрузку и служебные системы.



1.3

Полезной нагрузкой называют те устройства, которые выполняют главную задачу космического аппарата. Это камеры для спутников ДЗЗ, радиопередатчики для спутников связи или научное оборудование для исследовательских спутников.

В служебные системы входят все устройства, которые не относятся к полезной нагрузке. Их задачей является обеспечить корректную работу устройств полезной нагрузки. Как правило, к служебным системам относятся:

- Система управления (СУ). Представляет из себя управляющее устройство, т. н. бортовой компьютер. Отвечает за контроль состояния спутника и выполнение возложенных на него задач.

1.3 СОСТАВ КОСМИЧЕСКОГО АППАРАТА

- Система энергоснабжения (CЭС). Отвечает за обеспечение электрическим током всех устройств спутника, а также контролирует подаваемое на них напряжение и силу тока.
- Система связи (СС). Осуществляет передачу данных, полученных в ходе выполнения основных задач, т. е. передает фотографии или научную информацию.
- Система телеметрических измерений (СТМИ). Передает телеметрию – информацию, полученную от служебных систем: например, температуру на гранях спутника, входное напряжение на устройствах и пр. Данная информация необходима для анализа состояния спутника и контроля его работы, а также выявления причин поломок.
- Система ориентации и стабилизации (СОС). Отвечает за правильное положение спутника в пространстве.
- Система обеспечения теплового режима (СОТР). Обеспечивает обогрев или охлаждение спутника для поддержания необходимой температуры.

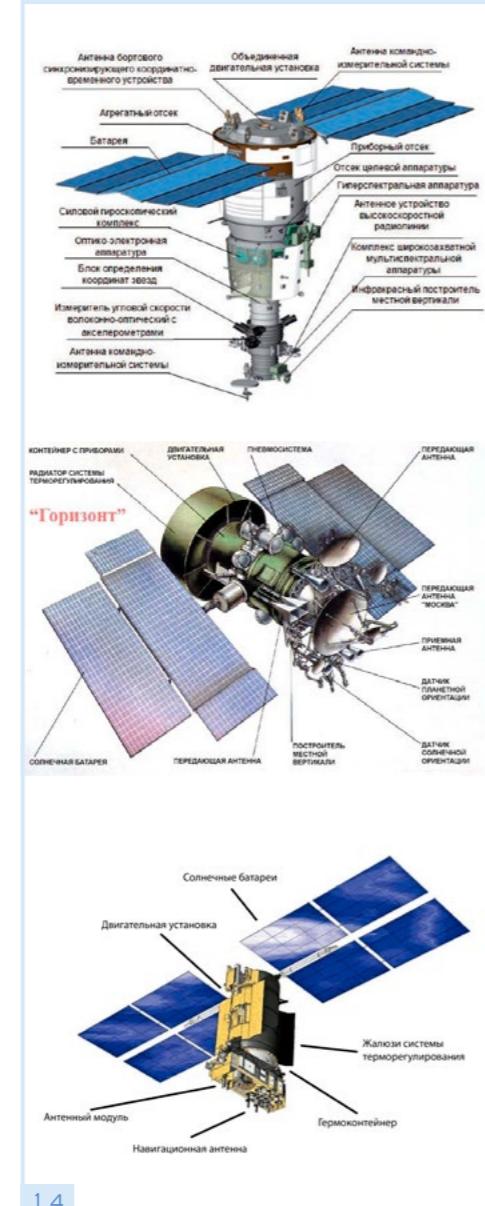
В зависимости от задачи, спутники могут сильно отличаться по конструкции и внешнему виду.

МАЛЫЕ КОСМИЧЕСКИЕ АППАРАТЫ

Прогресс в технических отраслях привел к миниатюризации многих устройств. Космические аппараты не являются исключением. Все чаще на смену большим спутникам приходят группировки из нескольких малых устройств.

СОВЕТ. Спросите учащихся, какого размера, по их мнению, бывают самые большие и самые маленькие спутники.

Одним из популярных форматов является т. н. кубсат (cubesat), представляющий из себя форму, близкую к кубу со стороной



1.4 ПРИМЕРЫ СПУТНИКОВ ДЗЗ, СВЯЗИ И НАВИГАЦИОННОГО

ТАБЛИЦА 1. КЛАССИФИКАЦИЯ СПУТНИКОВ ПО МАССЕ

Большие спутники	> 1000 кг
Малые спутники	500-1000 кг
Миниспутники	100-500 кг
Микроспутники	10-100 кг
Наноспутники	1-10 кг
Пикоспутники	< 1 кг
Фемтоспутники	< 100 г

Именно из-за своих малых габаритов и массы, кубсаты проще выводить на орбиту. Обычно эти наноспутники выводятся совместно с большими космическими аппаратами или запускаются в космическое пространство с борта МКС.

Объем корпуса, соответствующий кубу со стороной 10 см, называется юнитом и обозначается как 1U. Существуют несколько форматов кубсатов: 1U, 3U, 6U и т. п.

Кубсаты нашли свое применение в следующих областях:

- исследования в научной и инженерной областях при малых габаритах полезной нагрузки;
- отработка технических решений: если необходимо протестировать новую разработку, можно воспользоваться форматом наноспутника, чтобы в последующем использовать ее в более сложных по конструкции космических аппаратах;
- связи и ДЗЗ: небольшие спутники выводятся на низкие околоземные орбиты,

т. к. работа на относительно небольших высотах позволяет уложиться в установленный форматом объем;

- образовательные цели.

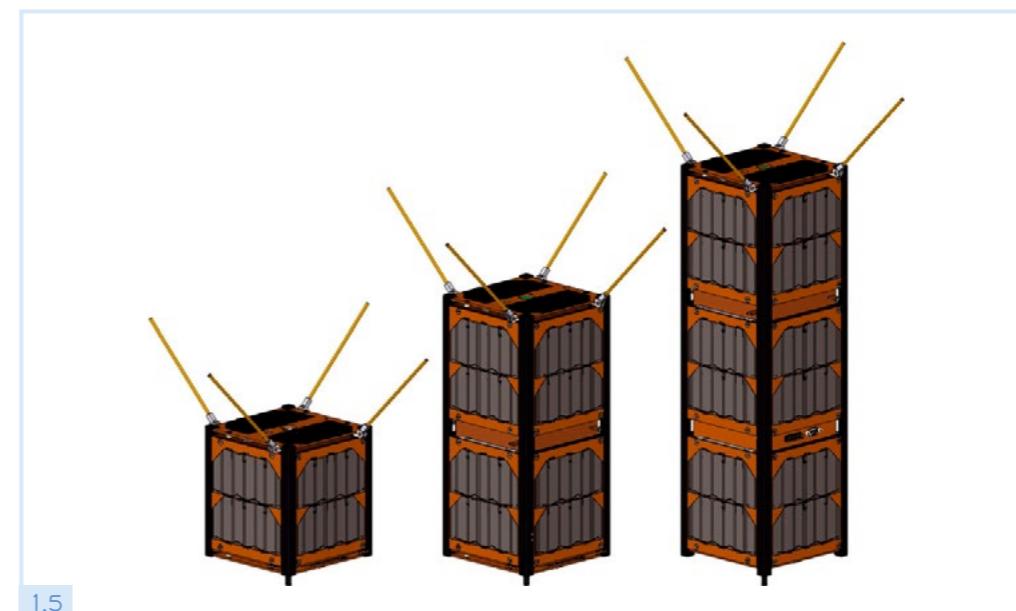
Малый объем кубсатов является как их преимуществом, так и недостатком. Не всегда есть возможность обеспечить кубсат полноценным набором подсистем так, чтобы это не сказалось на корректной работе.

ЧТО ДАЛЬШЕ?

Далее предстоит более углубленное изучение спутниковых систем на базе конструктора кубсата. На последующих занятиях предстоит сборка системы с поворотной батареей и конструирование простой системы ориентации и стабилизации.

Р. С.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru.



1.5 ФОРМАТЫ КУБСАТОВ 1U, 2U И 3U

СОВЕТ. Для лучшего знакомства с особенностями и задачами космической техники можно провести интерактивный урок НТИ по профилю «Системы связи и дистанционного зондирования Земли» (1 ак. час), материалы которого доступны по адресу:

<https://barvikha.odinedu.ru/documents/космическая%20игра.pdf>

Для проведения такого занятия учащимся понадобится только доступ в интернет.

ЧАСТЬ II

НЕБЕСНАЯ МЕХАНИКА

В ДАННОМ МОДУЛЕ РАССМАТРИВАЕТСЯ МЕХАНИКА ДВИЖЕНИЯ КОСМИЧЕСКИХ АППАРАТОВ, ИЗУЧАЮТСЯ ПРИНЦИПЫ ПОСТРОЕНИЯ РЕАЛЬНЫХ ОРБИТ ИСКУССТВЕННЫХ СПУТНИКОВ ЗЕМЛИ С ПОМОЩЬЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ GMAT И СИМУЛЯТОРА «ОРБИТА».

УЧАЩИЕСЯ ЗНАКОМЯТСЯ С ЗАКОНАМИ НЕБЕСНОЙ МЕХАНИКИ И УЧАТСЯ ПРОИЗВОДИТЬ НЕОБХОДИМЫЕ РАСЧЕТЫ ДЛЯ ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ ОРБИТ.

СИМУЛЯТОР «ОРБИТА» – ИГРОВОЙ ОБРАЗОВАТЕЛЬНЫЙ СЕРВИС, ПОЗВОЛЯЮЩИЙ МОДЕЛИРОВАТЬ РАБОТУ КОСМИЧЕСКИХ АППАРАТОВ НА ОРБИТЕ В УСЛОВИЯХ, БЛИЗКИХ К РЕАЛЬНЫМ. С ПОМОЩЬЮ СИМУЛЯТОРА УЧАЩИЕСЯ ПРОВЕРЯЮТ СВОИ ГИПОТЕЗЫ И ИЗУЧАЮТ РАБОТУ АППАРАТА НА ОРБИТЕ.

Теоретическая часть



ГРУППОВАЯ
РАБОТА



2 ЧАСА

13

Движение по орбите.

ЦЕЛЬ ЗАНЯТИЯ

В рамках занятия учащиеся знакомятся с основными принципами орбитального

движения, а также учатся подбирать орбиты с помощью ПО GMAT.

SОВЕТЫ УЧИТЕЛЮ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ. Первая часть занятия посвящена общей теории орбитального движения и включает в себя краткую справку по гравитационному взаимодействию и законам Кеплера, а также описывает особенности эллипса. Подачу теоретического материала желательно выстраивать в виде диалога с аудиторией.

Вторая часть занятия совмещает в себе теорию и практику: обучение работе в GMAT идет параллельно с подачей нового теоретического материала. Это способствует более легкому усвоению материала, так как теория иллюстрируется изображениями смоделированных орбит и треков спутников.

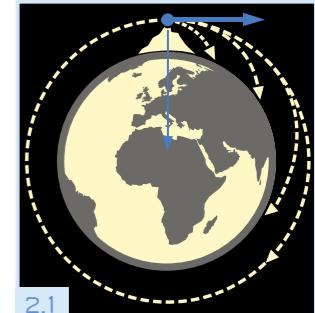
Преподаватель может демонстрировать учащимся работу в GMAT на экране своего ПК, выполняя указанные в данной части действия.

Программное обеспечение GMAT является бесплатным. Его можно скачать с сайта NASA:
<https://software.nasa.gov/software/GSC-17177-1>

ТЕОРИЯ К ПРАКТИКЕ

Как уже говорилось ранее, космические искусственные спутники Земли находят огромное количество применений. Причем многие из этих задач можно технически решать и без применения космических аппаратов: например, для фотосъемки привлекать авиацию, для радиосвязи – наземные ретрансляторы. Именно космические аппараты выгодно использовать для подобных целей благодаря особенностям их движения. Способность длительное время двигаться на нужном расстоянии от Земли с нужной скоростью почти без затрат топлива – именно та причина, по которой космические аппараты «отвоевали» многие области применения у наземных средств и авиации. Еще одна причина – уникальные физические условия: невесомость, отсутствие атмосферы и прочие факторы, сопутствующие открытому космосу.

Что же позволяет космическому аппарату длительное время двигаться вокруг Земли вопреки ее притяжению? Мысленно проведем эксперимент: поднимем спутник на высоту его орбиты (скажем, 600 км над Землей) и отпустим. Спутник упадет на Землю, двигаясь по прямой линии*. Если спутнику, перед тем – как сбросить, придать некоторую начальную скорость по касательной к земному шару, он упадет уже по криволинейной траектории. Если скорость увеличивать, точка падения будет удаляться все дальше и дальше, пока кривая не замкнется в круг (рисунок 2.1). Скорость, при которой это происходит, называется круговой $V_{kr}(r)$. Круговая скорость зависит от массы планеты, вокруг которой запускается спутник, и от высоты круговой орбиты. Круговая скорость, рассчитанная для точки на поверхности



2.1 ВЫВОД ТЕЛА НА ОРБИТУ
ПРИ ПОСТЕПЕННОМ
УВЕЛИЧЕНИИ СКОРОСТИ

* Если пренебречь движением Земли вокруг своей оси.

планеты (нулевая высота), называется первой космической V_{K1} . К примеру, первая космическая скорость для Земли равна 7 910 м/с, а круговая скорость для орбиты Марса высотой, скажем, 1000 км равна 3 546 м/с.

Иными словами, тело, вращающееся по орбите, «падает» так же, как и все остальные тела, но благодаря касательной скорости и кривизне земной поверхности это «падение» никогда не прекращается.

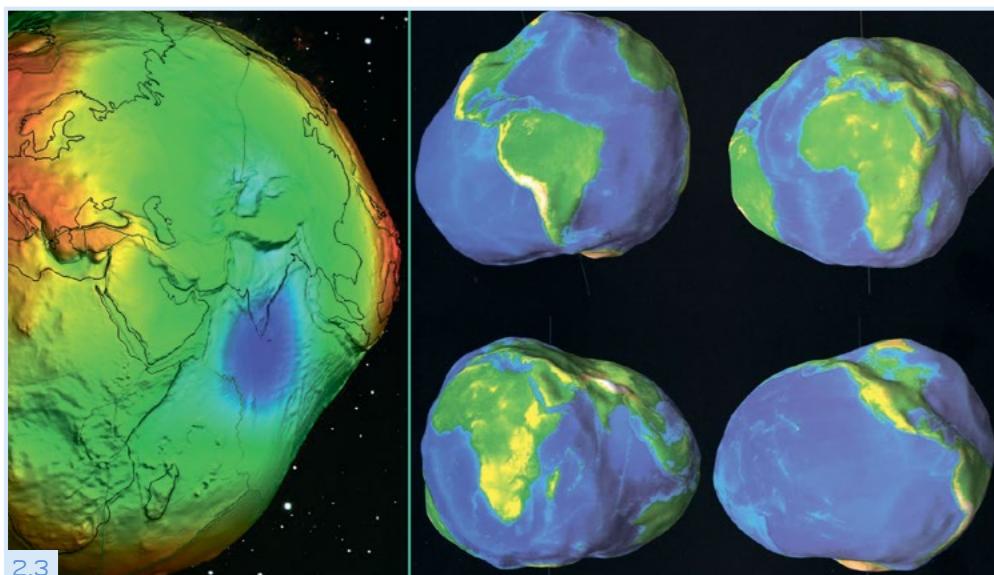
Как известно из школьного курса физики, тело совершает круговое движение под действием центростремительного ускорения. Оно направлено в сторону центра вращения. Для раскрученного тела, привязанного нитью к некоторой точке, это ускорение создается силой натяжения нити (рисунок 2.2).

$$a = \frac{V^2}{r} \quad (1)$$

В случае кругового движения спутника, естественно, никаких материальных нитей не существует. Центростремительное ускорение создается силой гравитации небесного тела.

Гравитация – явление, при котором любые два тела, обладающие массой, притягиваются друг к другу с равной для каждого тела силой, пропорциональной произведению их масс и обратно пропорциональной квадрату ускорения.

Еще в XVII веке Исаак Ньюton сформулировал закон, который позволяет



2.3 ФИГУРА ЗЕМЛИ (МАСШТАБ НЕРОВНОСТЕЙ УВЕЛИЧЕН)

расчитать ее величину и создаваемое ею ускорение:

$$F(r) = G \frac{Mm}{r^2} \quad (2)$$

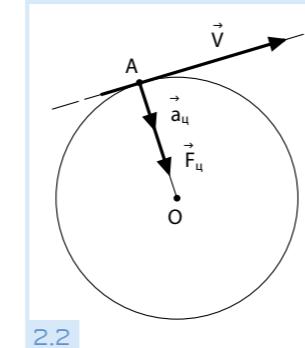
$$g(r) = \frac{GM}{r^2} = \frac{\mu}{r^2} \quad (3)$$

где M , кг – масса первого объекта (Земли или другого небесного тела),
 m , кг – масса второго тела (спутника),
 r , м – расстояние от центра масс Земли до объекта,
 G – коэффициент, называемый гравитационной постоянной Ньютона,

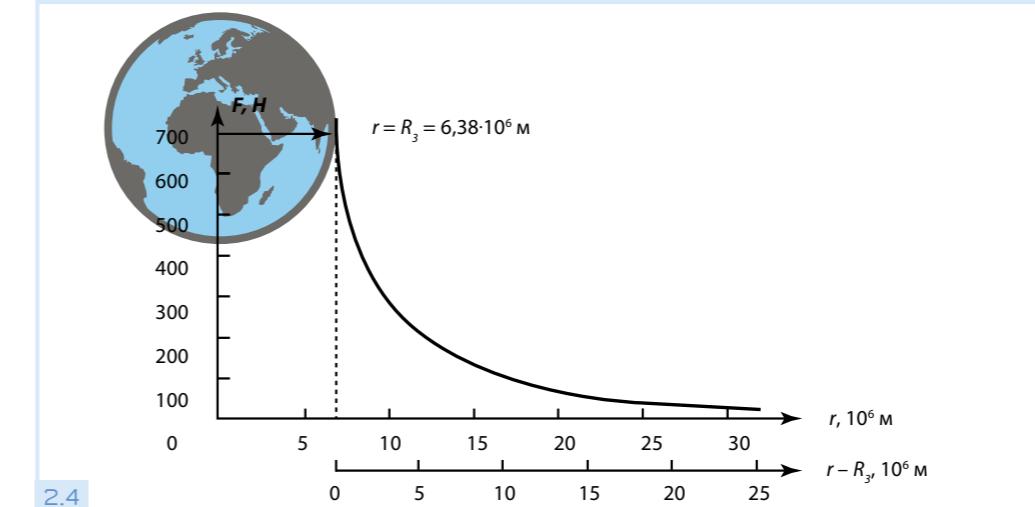
$$G = 6,67408 \cdot 10^{-11}, \text{Н} \cdot \frac{\text{м}^2}{\text{кг}^2}$$

μ , $\text{км}^3/\text{с}^2$ – гравитационный параметр небесного тела (вводится как произведение $\mu=GM$ для краткости записи), g – ускорение свободного падения, придаваемое телу силой тяжести.

Закон (2) идеально работает, если считать Землю идеально сферической и однородной и при приближенных вычислениях. Для более точных расчетов – например, прогноза орбит ИСЗ на длительный срок – нужно учитывать сплюснутость Земли на полюсах, а также другие неоднородности поля тяготения Земли.

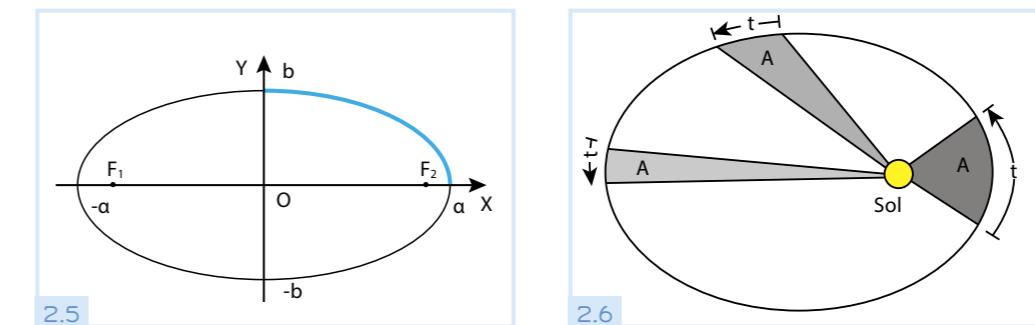


2.2 УСКОРЕНИЕ И СКОРОСТЬ ПРИ КРУГОВОМ ДВИЖЕНИИ

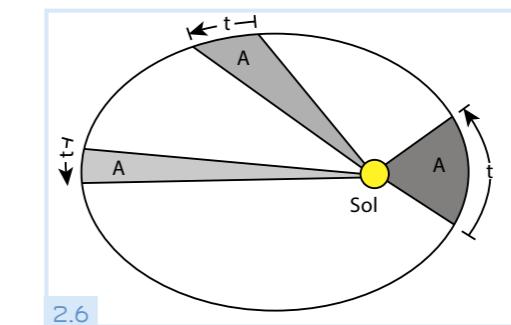


2.4

2.4 ГРАФИК ЗАВИСИМОСТИ УСКОРЕНИЯ СИЛЫ ТЯГОТЕНИЯ ОТ РАССТОЯНИЯ



2.5



2.6

2.5 ФОРМА ЭЛЛИПСА, ЕГО ФОКУСЫ И ПОЛУОСИ

2.6 ИЛЛЮСТРАЦИЯ ВТОРОГО ЗАКОНА КЕПЛЕРА

Если изобразить зависимость силы гравитации от расстояния, получится следующий график (рисунок 2.4). Зная формулы (1, 3), можно рассчитать круговую скорость для заданного радиуса и период орбиты T (время, за которое совершается полный виток):

$$\begin{aligned} V_{\text{кр}}(r) &= \sqrt{gr} = \\ &= \sqrt{\frac{\mu}{r^2} \cdot r} = \sqrt{\frac{\mu}{r}} \\ T(r) &= \frac{2\pi r}{V} = 2\pi r \sqrt{\frac{r}{\mu}} \end{aligned} \quad (4)$$

Соответственно, из (4) можем получить формулу для первой космической скорости:

$$V_{\text{K1}} = V_{\text{кр}}(r_0) = \sqrt{\frac{\mu}{r_0}} \quad (5)$$

где r_0 – условный радиус поверхности

Земли, примерно равный 6 371 км.

Чтобы в лабораторных условиях сымитировать орбитальное движение, нужно создать силу, обратно пропорциональную расстоянию до некоторой точки или оси.

Однако орбита вовсе не обязательно должна быть круговая. Немецкий ученый Иоганн Кеплер, живший в XVI-XVII веках, наблюдая движение планет вокруг Солнца, установил законы движения небесных тел, которые справедливы и для искусственных космических аппаратов: траектория движения объектов в гравитационном поле небесных тел имеет форму эллипса (рисунок 2.5).

Эллипс – фигура, координаты всех точек которой соответствуют уравнению

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (6)$$

где a и b – соответственно, большая и малая полуоси.

Как видно, если $a=b$, эллипс становится

кругом. При этом небесное тело, вокруг которого вращается объект, расположено в одной из двух точек, называемых **фокусами** эллипса. За равные промежутки времени линия, соединяющая объект с небесным телом, «кометает» равные площади (рисунок 2.6).

Для разных орбит, квадраты периодов обращения соотносятся как кубы больших полуосей:

$$\frac{T_1^2}{T_2^2} = \frac{a_1^3}{a_2^3}$$

(7)

Вытянутость эллипса характеризуется т.н. эксцентриситетом e :

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

(8)

Для круговых орбит $e=0$, чем ближе e к единице, тем более вытянута орбита. Если $e=1$, то орбита размыкается и превращается в параболу.

Точка орбиты, находящаяся ближе всего к притягивающему телу, называется **перицентром** и часто обозначается греческой буквой π . Перицентр околоземной орбиты называется перигеем (от имени греческой богини Земли Геи). Аналогично, точка наиболее удаленная – **апоценитр**, обозначается a . Апоцентр околоземной орбиты называется, соответственно, апогеем. Линия, соединяющая апоцентр и перицентр, называется **линией апсид**.

Учащимся можно дать задание найти в открытых источниках информации названия апоцентра и перицентра для спутников Солнца, Луны, Марса и, по возможности, других тел Солнечной системы.

Помимо законов Кеплера, для тел, движущихся по орбите, важным является закон сохранения энергии. Механическая энергия тела, вращающегося по орбите, заключена в потенциальной энергии сил гравитации и кинетической энергии; и их сумма не меняется:

$$E = E_{\text{п}} + E_{\text{к}} = \underbrace{-G \frac{Mm}{r}}_{\text{потенциальная}} + \underbrace{\frac{mV^2}{2}}_{\text{кинетическая}} = \text{const} < 0$$

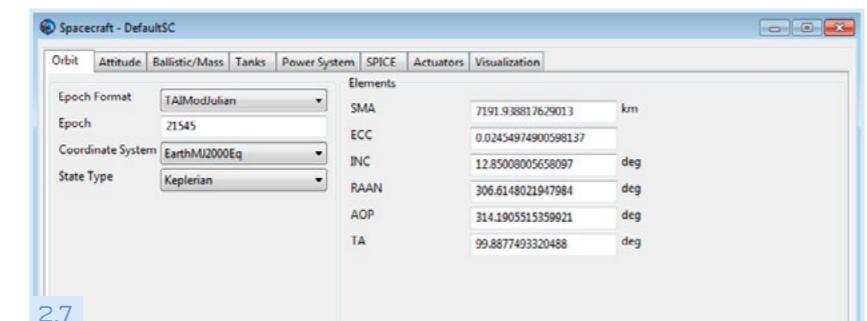
(9)

Из этого равенства следует важный вывод: в перицентре потенциальная энергия падает до минимума, а кинетическая возрастает, в апоцентре – наоборот. Иными словами, в перицентре орбиты скорость максимальная, а в апоцентре – минимальная.

Эти превращения энергии наглядно можно сравнить с нитяным маятником. В апоцентре, как и в самой верхней точке движения маятника с наибольшей потенциальной энергией, маятник имеет наименьшую скорость, а в перицентре, как и в самой нижней точке движения маятника – разгоняется до наибольшей скорости, но теряет потенциальную энергию, а сумма кинетической и потенциальной все это время остается постоянной.

НАЧАЛО РАБОТЫ В GMAT

Чтобы разобраться с тем, как происходит движение реальных космических аппаратов в околоземном пространстве, мы будем использовать GMAT (General Mission Analysis Tool) – программное обеспечение с открытым исходным кодом, разработанное NASA, обладающее широкими возможностями для моделирования различных космических миссий: от построения орбит искусственных спутников Земли до просчитывания гравитационных маневров для дальних космических миссий.



2.7

2.7 ВНЕШНИЙ ВИД ОКНА РЕДАКТИРОВАНИЯ ПАРАМЕТРОВ ОРБИТЫ

При запуске программы автоматически будет создан космический аппарат с настройками по умолчанию. Во вкладке Resources нужно выбрать наш единственный космический аппарат – Spacecraft -> DefaultSC. Его можно переименовать с помощью клавиши F2. По двойному нажатию мыши откроется окно настроек, где в первой вкладке Orbit можно произвести настройку параметров орбиты.

СИСТЕМА КООРДИНАТ

Для начала выберем координатную систему (Coordinate System). На выбор предоставляет несколько координатных систем, в том числе, эллиптическая и неподвижная относительно Земли. Но нам нужна система **EarthMJ2000Eq** – связанная с экватором Земли.

Эта плоскость отсчета удобна, так как наблюдения мы проводим с Земли, но на самом деле она обладает неприятной особенностью: координаты наблюдаемых тел относительно этой плоскости меняются с течением времени. Это происходит из-за прецессии земной оси: на небесной сфере ось описывает окружность с центром в северном полюсе эклиптики (для северного полушария) с угловым радиусом примерно 23,5°. По-другому это явление называют предварением равноденствий, так как при этом точки весеннего и осеннего равноденствия постоянно смещаются навстречу видимому годичному движению Солнца. Вместе с прецессией всегда происходит нутация – менее заметные, но всё равно важные для точных расчетов колебания наклона земной оси.

Поэтому, если мы хотим работать в системе координат, связанной с земным экватором, необходимо учитывать, как меняется с течением времени его положение относительно неподвижных далёких звёзд. Для этого все координаты, полученные в разное время и различными способами, должны быть приведены к единой системе координат.

В 1998 году Международный астрономи-

ческий союз (МАС) решил, что эта система координат, называемая небесной системой отсчета, будет определяться координатами внегалактических радиоисточников. Для удобства МАС также рекомендуют, чтобы основная плоскость этой системы отсчета была как можно ближе к небесному экватору для выбранного момента времени – полдня 1 января 2000 года. Этот момент времени, согласованный астрономами со всего мира, называется стандартной эпохой.

ЭПОХА

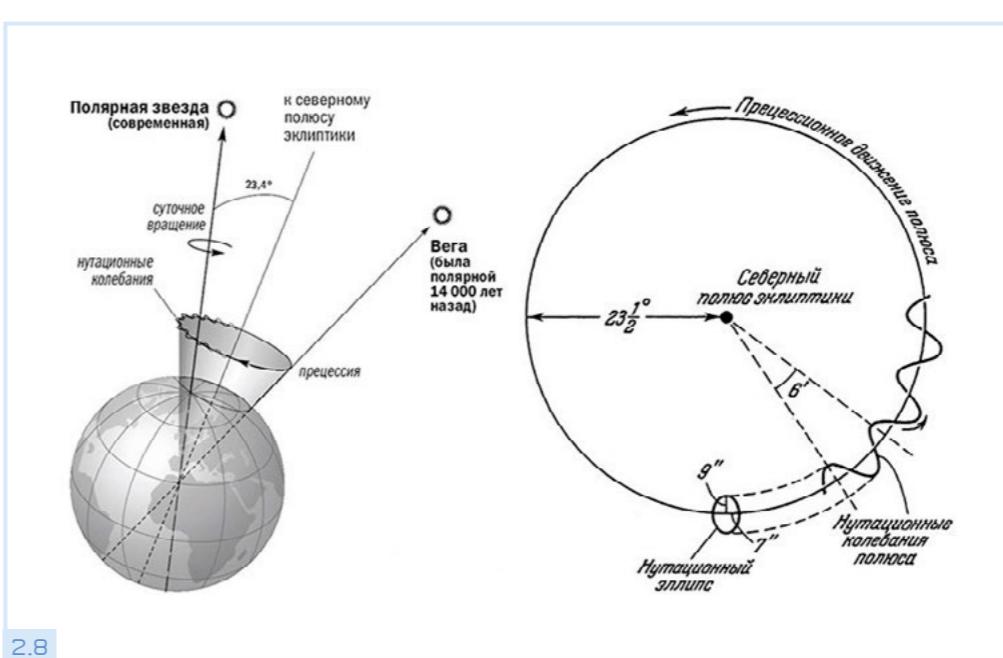
Эпоха в астрономии – момент времени, для которого определены координаты или элементы орбиты. Момент проведения наблюдения называют эпохой наблюдения.

Стандартная эпоха обозначается как J2000.0. Переведем её в систему юлианских дат. Юлианская дата (JD) отсчитывается от 1 января 4713 до н. э. по юлианскому календарю. Длительность юлианского дня равна 86 400 секундам, если относить юлианскую дату к земному времени. Тогда стандартная эпоха в системе юлианских дат будет равна 2 451 545,0.

Часто удобнее использовать модифицированную юлианскую дату MJD, рекомендованную МАС: $MJD = JD - 2\,400\,000,5$.

Стандартная эпоха по международному атомному времени (TAI) – 11:59:27,816 1 января 2000 года.

Стандартная эпоха по всемирному координированному времени (UTC) – 11:58:55,816 1 января 2000 года.



2.8 ПРЕЦЕССИЯ И НУТАЦИЯ ЗЕМНОЙ ОСИ

В GMAT эпоху можно задавать в различных форматах (epoch format). Если вы хотите использовать форматы юлианской даты, обратите внимание, что в GMAT используется модифицированная юлианская дата (ModJulian), которая связана с юлианской датой (JD) следующим соотношением:

$$\text{ModJulian} = \text{JD} - 2430000.0$$

Таким образом, стандартная эпоха J2000, приведенная к модифицированной юлианской дате, будет равна:

$$\text{ModJulian} = 2451545.0 - 2430000.0 = 21545$$

Чтобы привести эпоху из данных TLE к модифицированной юлианской дате, можно использовать формулу:

$$\text{ModJulian} = 21545 + (\text{год эпохи}) * 365.25 + (\text{время эпохи})$$

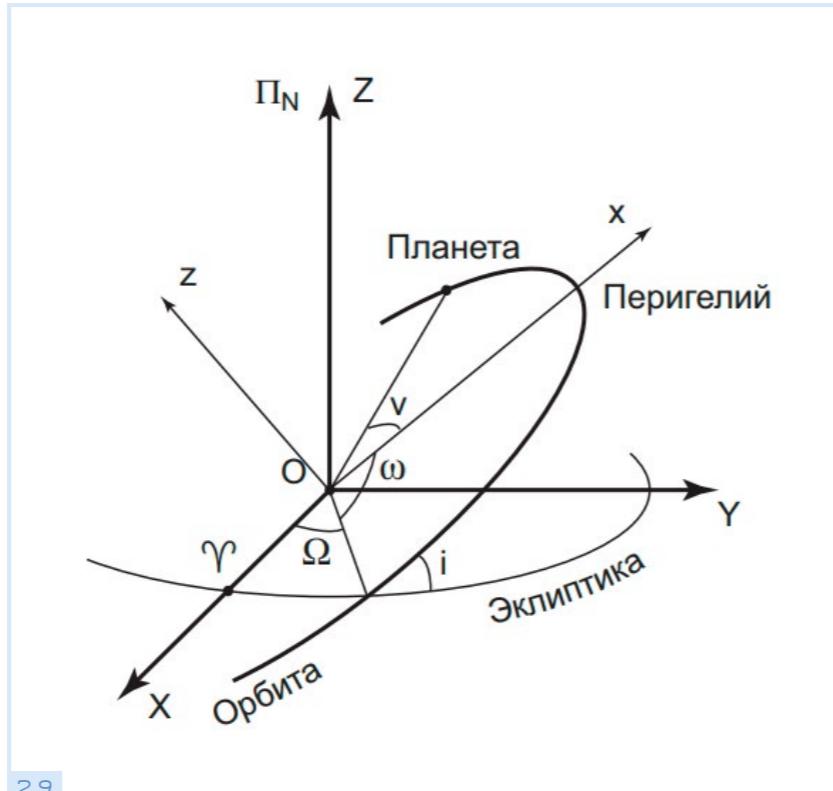
Также в GMAT можно использовать григорианскую дату в формате dd mmm yyyy HH:MM:SS.FFF, где dd – дни, mmm – месяц (первые три буквы по-английски), yyyy – год, hh – часы, mm – минуты, ss – секунды, fff – дробная часть секунд. В таком случае придётся переводить время эпохи в месяцы, дни, часы, минуты и секунды.

СПОСОБ ЗАДАНИЯ ЭЛЕМЕНТОВ ОРБИТЫ

Далее следует выбрать способ задания элементов орбиты (State Type). Однозначно задавать положение аппарата на орбите можно различными способами: например, через привычные прямоугольные координаты, когда задаются непосредственно координаты аппарата по осям X, Y и Z и компоненты его скорости по этим осям, причём оси X и Y лежат в плоскости земного экватора. Также можно выбрать сферические, эллиптические координаты и многие другие, но мы будем использовать кеплеровы элементы орбиты, так как они дают более интуитивно понятное представление о форме орбиты и расположении её относительно земного экватора.

КЕПЛЕРОВЫ ЭЛЕМЕНТЫ ОРБИТЫ

Кеплеровы элементы включают в себя шесть элементов орбиты: большую полуось, эксцентриситет, наклонение, долготу



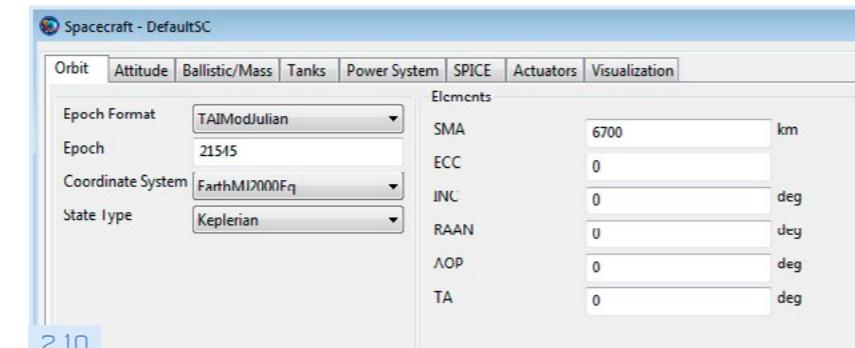
2.9

2.9 КЕПЛЕРОВЫ ЭЛЕМЕНТЫ ОРБИТЫ

восходящего узла, аргумент перицентра и аномалию. Рассмотрим каждый из них по отдельности.

1. a – большая полуось. Соединяет перигей и апогей аппарата. Проходит через центр Земли.

В программе GMAT в окне настройки орбиты большая полуось обозначается как SMA – semi-major axis. Построим несколько орбит с различными значениями большой полуоси. Для этого обнулим все поля элементов орбиты, кроме SMA:



2.10

2.10 НАСТРОЙКА ЭЛЕМЕНТОВ ОРБИТЫ

В поле SMA запишите значение 6700 км – оно соответствует высоте около 330 км над поверхностью Земли. Так как все другие элементы орбиты равны нулю, мы получим круговую орбиту, лежащую в плоскости земного экватора. После того, как все поля будут заполнены, нужно «применить» изменения и закрыть окно настроек.

Запустите моделирование, оставив все другие параметры моделирования «по умолчанию». Для этого необходимо нажать F5 или синюю стрелку Run. Появится два окна, в которых можно наблюдать, как просчитываются орбита и трек аппарата. С помощью клавиши F9 можно запустить анимацию и наблюдать за перемещением аппарата медленнее или быстрее, используя кнопки с двойными стрелками – Faster Animation and Slower Animation.

Появятся два окна: в одном отображается орбита космического аппарата, в другом – его трек, показывающий, как меняются широта и долгота, над которыми пролетает спутник. Постройте орбиты с большой полуосью 10 000 км; 20 000 км. Наблюдайте за изменениями. Орбита с большой полуосью не замкнута, потому что времени моделирования, выставленного по умолчанию, не хватает для того, чтобы спутник совершил полный оборот вокруг Земли.

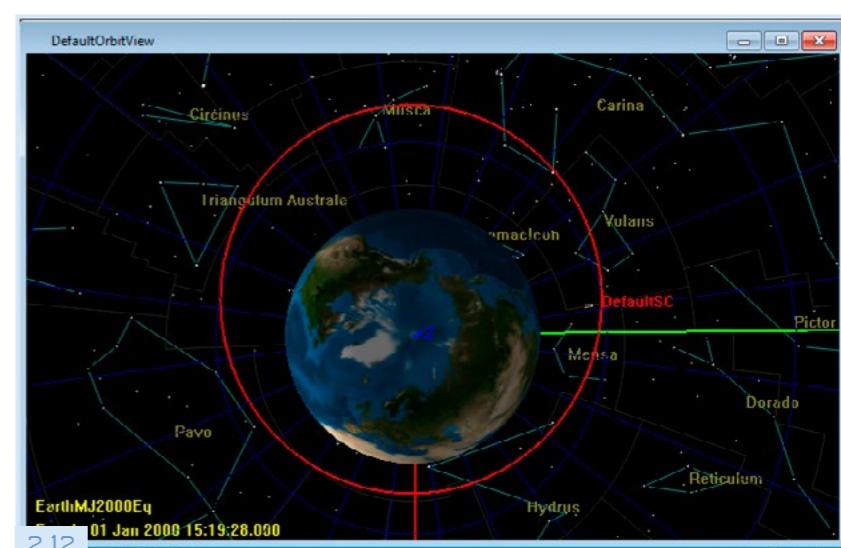
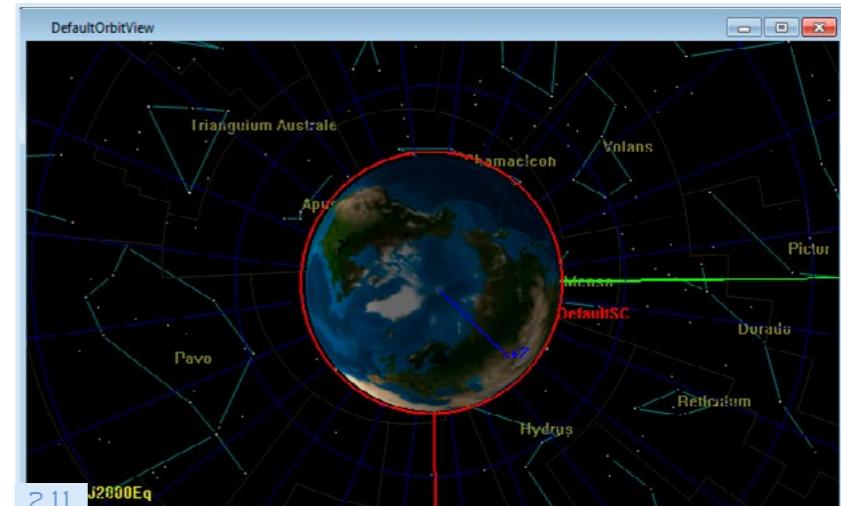
СОВЕТ. Чтобы приблизить или отдалить изображение, перемещайте мышь с зажатой правой кнопкой. Чтобы повернуть плоскость орбиты относительно плоскости экрана, перемещайте мышь с зажатой левой кнопкой. Чтобы повернуть изображения в плоскости экрана, перемещайте мышь с зажатым колесиком. Чтобы сместить точку взора, перемещайте мышь с зажатой левой кнопкой и клавишей Shift.

2. e – эксцентриситет. Чем больше эксцентриситет, тем более вытянутая орбита:

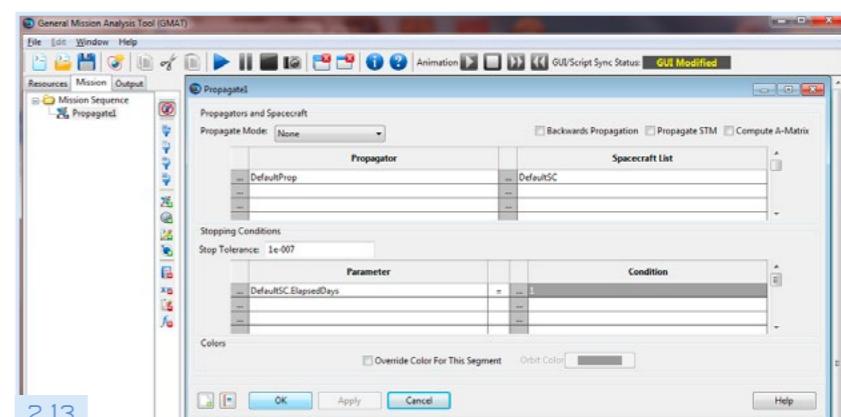
- при $e=0$ орбита является окружностью
- при $e>0$ орбита эллиптическая
- при $e=1$ орбита становится параболической
- при $e>1$ орбита гиперболическая (траектория межпланетного полёта)

В программе GMAT эксцентриситет обозначается как ECC – eccentricity. Откройте окно настроек орбиты и задайте значение большой полуоси 10 000 км, а эксцентриситет сделайте равным 0.2. Запустите моделирование и разверните отображение орбиты таким образом, чтобы координатная ось Z смотрела «на нас». Можно наблюдать, что орбита стала более вытянутой, а Земля больше не находится в её центре.

Для следующих экспериментов с орбитой понадобится изменить время моделирования, чтобы аппарат успевал сделать хотя бы один оборот вокруг Земли. Нужно задать конечное условие, при выполнении которого моделирование прекратится.



Для этого в GMAT нужно найти вкладку Mission и выбрать там Mission Sequence -> Propagate1. Это автоматически созданная, пока что единственная миссия. В конечных условиях (Stopping Conditions), нажав на три точки в начале строки, в таблице можно выбрать условие, по которому вы хотите завершить моделирование. Например, вместо стандартных ElapsedSeconds можно выбрать ElapsedDays и прописать в правой колонке подходящее число. Нам хватит одного дня.



2.11 ПРОСТЕЙШАЯ ОРБИТА С БОЛЬШОЙ ПОЛУОСЬЮ 6700 КМ

2.12 ЭЛЛИПСОВИДНАЯ ОРБИТА С ЭКСЦЕНТРИСИТЕТОМ 0.2

2.13 ОКНО НАСТРОЙКИ МИССИИ

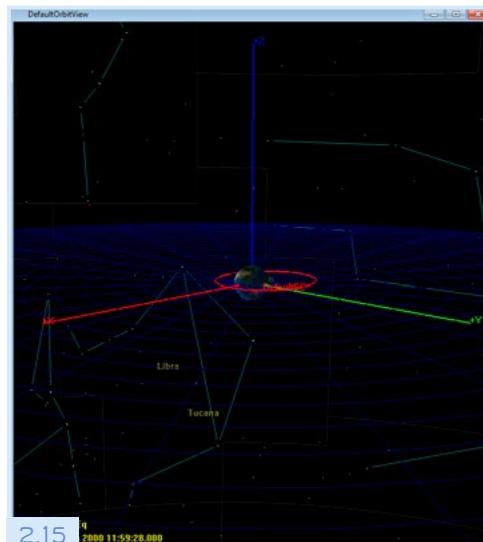
Нажмите «применить» и вернитесь во вкладку Resources к настройке орбиты. Поменяйте значение большой полуоси на 20000 км и постройте орбиты с эксцентриситетами 0.4; 0.65.

Как можно видеть, чем ближе эксцентриситет к единице, тем больше вытягивается эллипс и тем сильнее фокус, в котором находится центр масс Земли, смещается от центра.

3. i – наклонение – это угол между плоскостью орбиты и плоскостью отсчета:

- если $0^\circ < i < 90^\circ$, то движение небесного тела называется прямым, а орбита – восточной
- если $90^\circ < i < 180^\circ$, то движение небесного тела называется обратным, а орбита – западной

В GMAT наклонение обозначается как INC – inclination. Чтобы наглядно увидеть, как меняется орбита при изменении наклонения задайте значение большой полуоси 20 000 км, а эксцентриситет – 0.4. Оставьте наклонение равным 0 и запустите моделирование. Обратите внимание, что на изображении, где построена орбита, есть сетка синего цвета. Это плоскость XY, которая является плоскостью земного экватора. При наклонении орбиты, равном нулю, орбита лежит в этой плоскости:

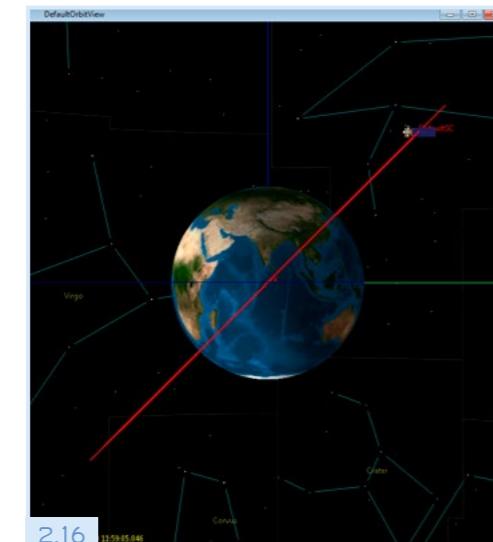


Теперь постройте орбиты с наклонениями 45° и 90° . Обратите внимание, как при этом меняется угол между плоскостью орбиты и плоскостью экватора.

4. Ω – долгота восходящего узла – это угол между направлением на точку весеннего



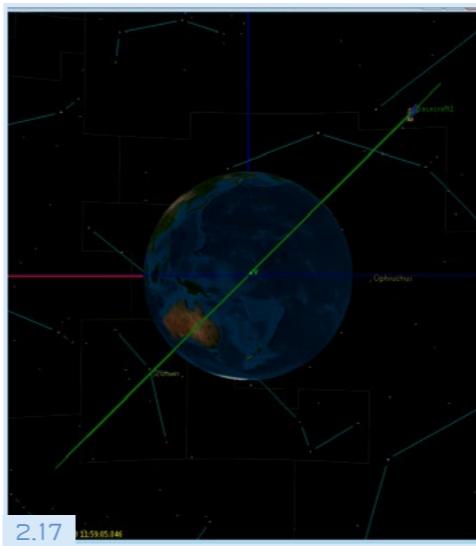
2.14 ОРБИТА С ЭКСЦЕНТРИСИТЕТОМ 0.65



2.15 ПЛОСКОСТЬ ХУ – ПЛОСКОСТЬ ЗЕМНОГО ЭКВАТОРА.

2.16 ВОСХОДЯЩИЙ УЗЕЛ ЛЕЖИТ НА ОСИ Х ПРИ $\Omega = 0^\circ$

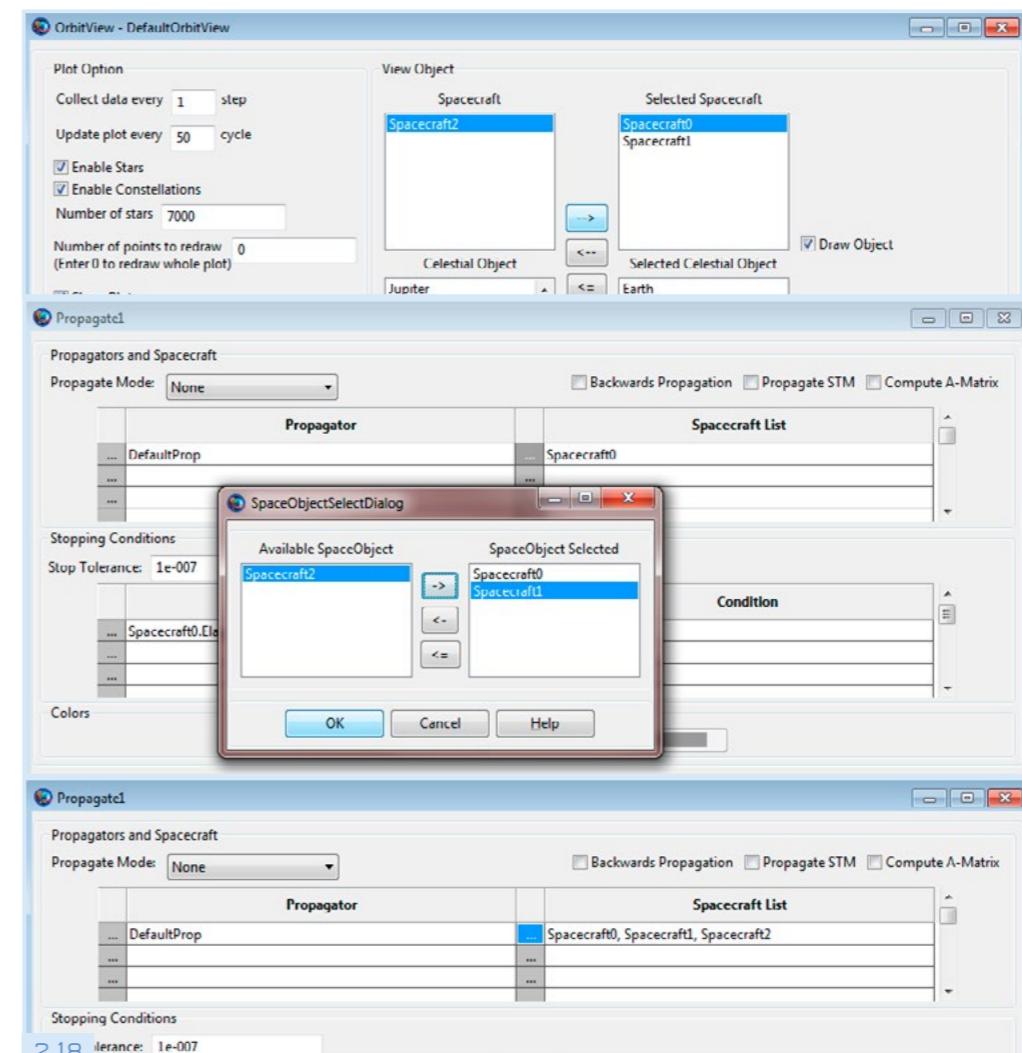
Теперь задайте долготе восходящего узла значения 90° ; 180° ; 270° . Система координат неподвижна, поэтому, по взаимному расположению восходящего узла и осей X и Y, можно понять, что орбита поворачивается вокруг оси Z. Например, для долготы 90° восходящий узел лежит на оси Y.



2.17

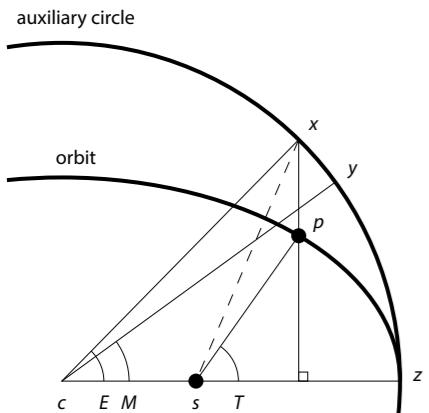
В GMAT аргумент перицентра обозначается как AOP – argument of periapsis. Для того чтобы наглядно продемонстрировать, как меняется орбита, создадим и настроим орбиты сразу трёх спутников. Для того чтобы добавить новый космический аппарат, нужно нажать правой кнопкой мыши на раздел Spacecraft и добавить новый корабль (Add Spacecraft). Задайте для всех трёх КА следующие значения: SMA=20 000 км, ECC=0.4, INC=45°, RAAN=0, TA=0. Аргументы перицентра для них задайте разные: например, 0° , 120° и 240° .

Затем зайдите в настройки отображения орбиты Output -> DefaultOrbitView и в разделе View Object с помощью стрелки переместите все аппараты из колонки Spacecraft в колонку Selected Spacecraft, чтобы они отобразились в ней. После чего зайдите во вкладку Mission, откройте настройки миссии Propagator1 и в разделе Propagators and Spacecraft добавьте в таблицу новые космические аппараты. Для этого нужно нажать на три точки в правой колонке и выбрать новые корабли.



2.17 ВОСХОДЯЩИЙ УЗЕЛ ЛЕЖИТ НА ОСИ У ПРИ $\Omega = 90^\circ$

2.18 ДОБАВЛЕНИЕ НОВЫХ КОСМИЧЕСКИХ АППАРАТОВ



Угол T – истинная аномалия, **угол M** – средняя аномалия, **точка p** – истинное положение тела; **точка s** – центральное тело, совпадающее с фокусом орбиты; **точка c** – центр орбиты и воображаемой окружности, по которой КА двигался бы с постоянной угловой скоростью; **точка z** –periцент, через который проходит орбита, и воображаемая окружность.

2.19

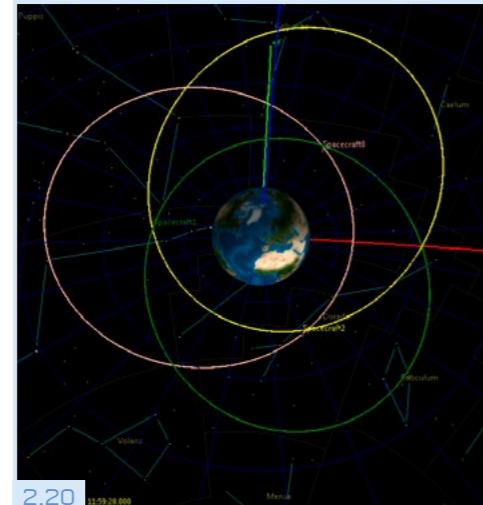
Запустите моделирование. В окне отображения орбит будут построены сразу три орбиты. Они лежат в одной плоскости; и линия узлов в данном случае всегда лежит на оси X, а линия апсид меняет свой угол по отношению к ней.

6. Аномалия:

- M_0 – средняя аномалия – это угловое расстояние от перигея орбиты до воображаемого положения космического аппарата, в случае, если он движется с постоянной угловой скоростью (см. рис. 2.19).
- $T_{(v)}$ – истинная аномалия. Представляет собой угол между радиус-вектором тела и направлением на перигея.

В GMAT есть возможность задавать только истинную аномалию – TA – true anomaly. Она нужна для того, чтобы обозначить, в какой именно точке орбиты находился аппарат на момент запуска моделирования. Задайте всем трём КА значение большой полуоси 30000 км, эксцентриситет – 0.65, наклонение, долготу и аргумент перигея 0, а истинные аномалии сделайте равными 0° , 180° и 210° . В настройках миссии снова задайте условие, по которому вы хотите завершить моделирование: верните параметр ElapsedSeconds и пропишите для него значение 5000 секунд.

Запустите моделирование. Выставленного времени недостаточно, чтобы аппарат сде-



2.19 РАЗИЦА МЕЖДУ СРЕДНЕЙ И ИСТИННОЙ АНОМАЛИЕЙ

2.20 ТРИ ОРБИТЫ С РАЗЛИЧНЫМИ АРГУМЕНТАМИ ПЕРИЦЕНТРА

2.21 ОДНА И ТА ЖЕ ОРБИТА С РАЗЛИЧНЫМИ ИСТИННЫМИ АНОМАЛИЯМИ



2.20

2.21

Практика II



ГРУППОВАЯ РАБОТА



2 ЧАСА

Расчет кеплеровых элементов орбиты космического аппарата по заданным условиям

ЦЕЛЬ ЗАНЯТИЯ

Познакомившись с элементами орбит, можно теперь решить задачу по подбору орбиты для заданных условий.

СОВЕТЫ УЧИТЕЛЮ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ.

Первые 15 минут занятия следует уделить проверке усвоения предыдущего материала: попросите учащихся сформулировать известные им законы небесной механики, назвать кеплеровы элементы орбиты, сделать поясняющие чертежи.

Для работы с симулятором «Орбита» необходим только доступ в Интернет. Доступ к задачам осуществляется по ссылке: <https://nti.orbitagame.ru/>. Для доступа к задачам необходимо зарегистрироваться на сайте. Подробный теоретический разбор задач из этой и следующей части практикума можно найти по ссылке:

<https://drive.google.com/drive/u/0/folders/1zse7y07PZJcIp82gumaT7A0xr3oJ-5WX>.

ТЕОРЕТИЧЕСКИЙ РАСЧЕТ

Разберем решение простой задачи.

Наноспутник формата 3U массой 2,96 кг может быть размещен на произвольной орбите 15 января 2020 года в 18:00 UTC. Требуется задать параметры орбиты через кеплеровы элементы так, чтобы 16-го, 17-го и 18-го января в 9:00 UTC каждого дня аппарат прошел над Норильском. Данные координаты Норильска, а также радиус Земли, гравитационный параметр (т. е., произведение гравитационной постоянной на массу Земли), средняя скорость вращения Земли и начальный угол вращения Земли.

- Координаты Норильска: 69.3535 по широте, 88.2027 по долготе.

- Радиус Земли, м: 6371008.8

- Гравитационный параметр, $\mu = G \cdot M$, м 3 /с 2 : 3.986004418·10 14

- Средняя скорость вращения Земли, об/сут: 1.00273781191135448

- Начальный угол вращения Земли: 24.66°

ВАЖНОЕ ПРИМЕЧАНИЕ

При расчете орбиты мы будем учитывать, что орбита не подвержена прецессии и отсутствует влияние атмосферы на движение космического аппарата. Также предположим, что плоскость отсчета совпадает с плоскостью Земного экватора.

- СОВЕТ. Дайте учащимся несколько минут, чтобы самостоятельно подумать над задачей, и обсудите с ними их идеи. Следующие выкладки и наводящие вопросы являются подсказками, которые помогут ученикам прийти к верному решению.

Задача дает нам свободу: мы сами можем выбирать удобный для дальнейших расчетов тип орбиты. Известно, что спутник должен несколько раз пролететь над определенной широтой и долготой, а орбита не подвержена прецессии. Как может быть ориентирована орбита в пространстве и какую форму она может иметь, чтобы максимально упростить задачу?

- Проще всего выбрать круговую орбиту ($e = 0$).
- Орбиту можно сделать полярной, чтобы

РЕЗУЛЬТАТ

Учащиеся освоили основы работы в программном обеспечении GMAT, получили наглядное представление о кеплеровых элементах орбит.

ЧТО ДАЛЬШЕ?

Теперь учащиеся могут приступить к моделированию орбит и орбитальных маневров, близких к реальным орбитам существующих ИСЗ.

P. S.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru.

спутник проходил над всеми широтами, а долгота при этом будет неизменной и равной долготе восходящего узла ($i = 90^\circ$).

3. Аргумент перицентра в случае круговой орбиты можно выбрать любым; проще сделать его таким, чтобыperiцентр находился на экваторе ($\omega = 0$).

4. Координаты спутника отсчитываются в экваториальной системе небесных координат. Каким образом можно связать её с земной системой отсчёта, чтобы узнать широту и долготу, над которыми пролетает спутник?

5. Ось X земной системы отсчёта лежит в плоскости нулевого меридиана. Начальный угол вращения – это угол между направлениями на нулевой меридиан Земли и на точку весеннего равноденствия.

6. Вычислите долготу восходящего узла спутника, исходя из долготы точки, над которой он должен оказаться, и учитывая вращение Земли.

7. Можно вычислить большую полуось орбиты, зная период. Какой период обращения может быть у спутника, который должен оказываться над одной и той же точкой в одно и то же время несколько дней подряд? Подсказка: для точных вычислений необходимо использовать звёздные сутки.

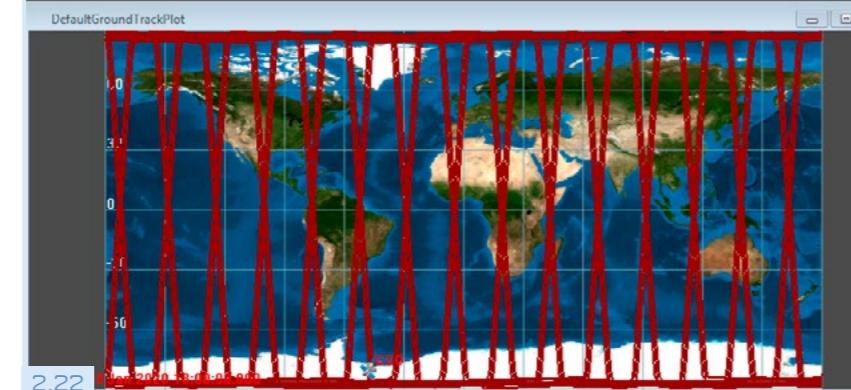
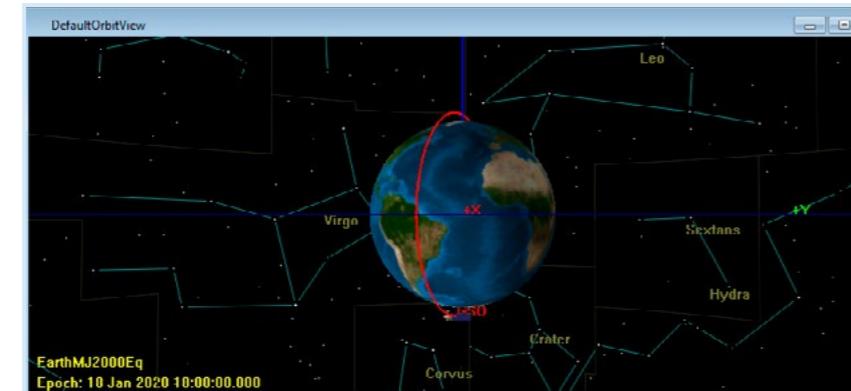
8. Аномалия отсчитывается от перицентра. В какой точке орбиты спутник должен начать движение, чтобы очутиться в заданное время над заданной широтой?

ПРОВЕРКА РЕШЕНИЯ В ПО GMAT

Постарайтесь выяснить, удовлетворяют ли рассчитанные элементы орбиты условию задачи.

Введите рассчитанные параметры в GMAT, отключите влияние гравитации Солнца и Луны, атмосферу и давление солнечной радиации (в нашей задаче спутник находится в идеальных условиях, не нужно даже учитывать прецессию и нутацию). Эпоха должна совпадать со временем размещения спутника на орбите, а остановить моделирование можно после последней контрольной точки (18 января в 9:00). Запустите моделирование.

Давайте попробуем проанализировать получившиеся орбиту и трек. Спутник совершает 15 оборотов за сутки, и на изображении все они накладываются друг на

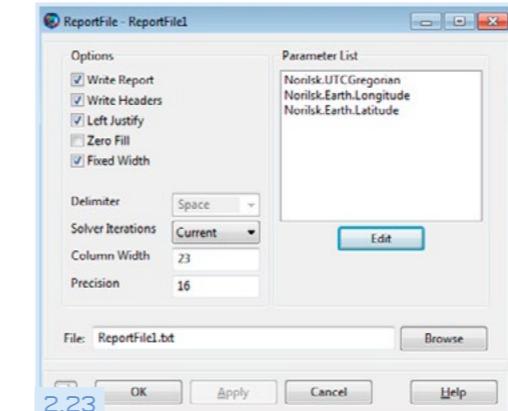


друга. Конечно, можно запустить замедленную анимацию и попытаться рассмотреть, над какой примерно точкой пролетал спутник в 9 утра, но можно получить точные результаты более удобным способом.

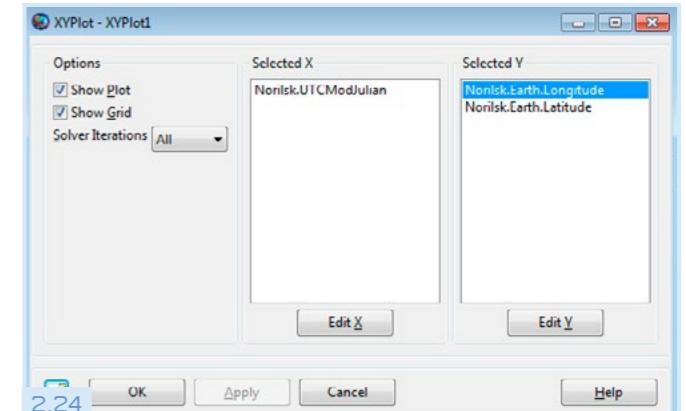
Найдите во вкладке *Resources* раздел *Output*. При нажатии на него правой кнопкой мыши появится возможность добавить новые выходные файлы, которые должны быть созданы при моделировании. Добавим отчет *ReportFile* и график *XYPlot* для наглядности. Чтобы понять, приблизились ли мы к верному решению задачи, необходимо узнать широту и долготу точки, над которой пролетал спутник 16, 17 и 18 января в 9:00. Настроим отображение отчета и графика соответствующим образом.

В настройках отчета необходимо отредактировать список параметров, которые будут в него внесены. Удалите все записанные по умолчанию параметры и добавьте дату и время в формате *UTCGregorian*, долготу *Longitude* и широту *Latitude*. В настройках графиков можно задать отображение того, как менялись долгота и широта от времени: за ось X необходимо принять эпоху в формате *UTCModJulian* (привычный григорианский формат здесь, к сожалению, недоступен), а за ось Y – долготу и широту.

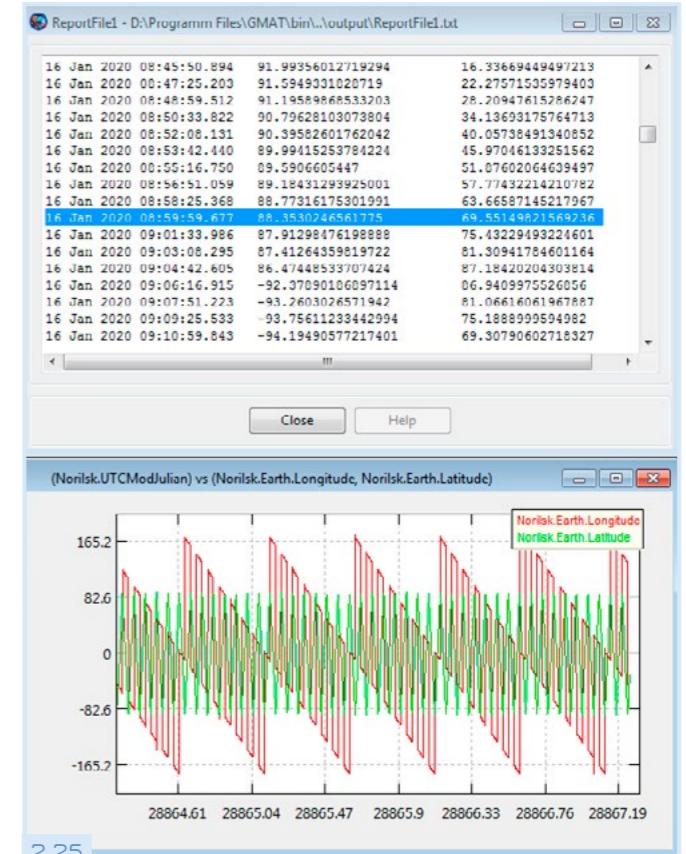
Снова запустите моделирование. Рядом с окнами отображения орбиты и трека



2.23



2.24



2.25

появится третье окно, с графиками долготы и широты от времени. Зайдите во вкладку *Output->Reports* и откройте отчет. Найдите долготу и широту, соответствующие времени 9:00.

Как видно из отчета, 16 января в 9 утра спутник находился над долготой 88.3530 и широтой 69.5515 (требуемые координаты – 88.2077 по долготе и 69.3535 по широте). Данный результат можно считать удовлетворительным и приступить к «полевым испытаниям» – проверке правильности решения в симуляторе «Орбита».

S Предложите учащимся поразмышлять, из-за чего координаты, полученные с помощью моделирования, отличаются от требуемых и соответствующих расчётом.

РАБОТА В СИМУЛЯТОРЕ «ОРБИТА»

На сайте nti.orbitagame.ru зайдите в событие «Подготовка к профилю «Спутниковые системы» Олимпиады КД НТИ» и выберите задачу «Аппарат над заданной точкой».

Выберите вкладку *Решения* и создайте новое решение, вписав в соответствующие поля рассчитанные значения элемент-

ов орбиты. Нажмите кнопку *Сохранить* сверху страницы, а затем – *Отправить решение*.

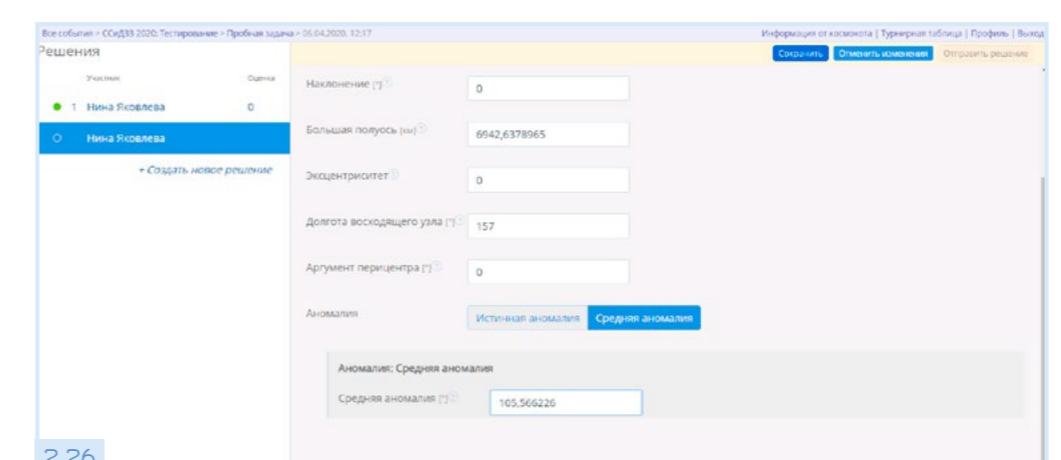
2.26

2.23 НАСТРОЙКИ ОТЧЕТА

2.24 НАСТРОЙКИ ГРАФИКОВ

2.25 ЗАВИСИМОСТЬ ДОЛГОТОЫ И ШИРОТОЫ, НАД КОТОРЫМИ ДВИЖЕТСЯ СПУТНИК, ОТ ВРЕМЕНИ

2.26 СОЗДАНИЕ НОВОГО РЕШЕНИЯ В «ОРБИТЕ»



2.26

Новое решение добавляется в конец списка слева. Как только решение задачи будет отправлено, кружок рядом с именем участника изменит цвет на красный, а когда результат будет известен – на зелёный. Нажмите на кнопку *Результаты*, чтобы узнать, сколько очков успеха заработка.

Если задача решена неверно или неудовлетворительно, проанализируйте графики. Первый график показывает, как менялись с течением времени широта и долгота, над которыми пролетал аппарат, на втором графике можно увидеть его высоту. Каждый график можно масштабировать колесом мыши. На последнем рисунке можно посмотреть трек спутника.

РЕЗУЛЬТАТ

Учащиеся научились рассчитывать элементы орбиты по заданному условию, глубже освоили инструментарий GMAT. Научились работать в симуляторе «Орбита»,

который используется на втором туре олимпиады НТИ по профилю «Спутниковые системы».

ЧТО ДАЛЬШЕ?

Задача, решенная в этом разделе была пробной и решение подробно разбиралось. Если вам понравилось изучение орбитальной механики, то можно продолжить занятия по методическому пособию «Небесная механика» или решать задачи в доступном событии симулятора «Орбита».

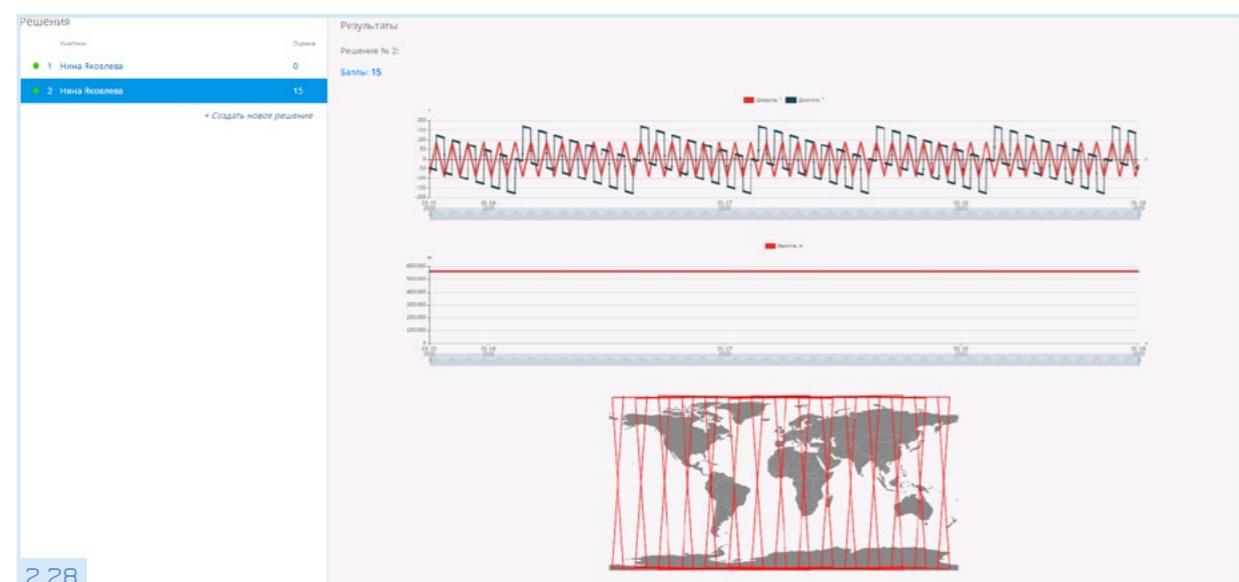
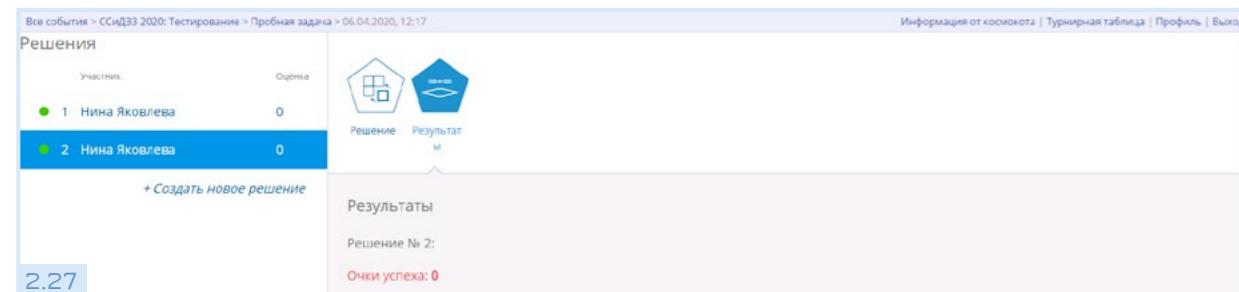
Если же хочется побыстрее перейти к работе с «железом», то в следующем разделе вам предстоит первое знакомство с конструктором IntroSat.

P. S.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru.

2.27 СПИСОК СОЗДАННЫХ РЕШЕНИЙ И ОКНО РЕЗУЛЬТАТОВ

2.28 РЕЗУЛЬТАТЫ МИССИИ



ЧАСТЬ III

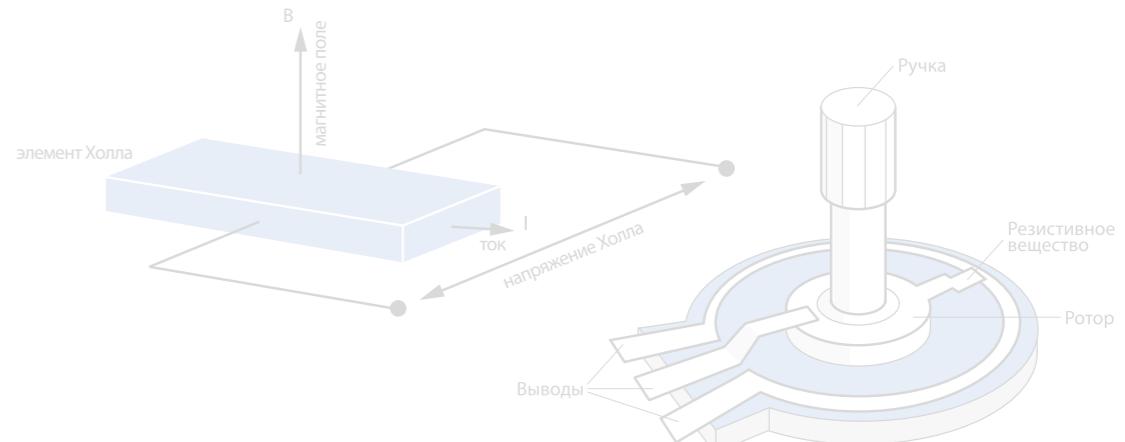
ДАТЧИКИ НА СПУТНИКЕ

МОДУЛЬ ПОСВЯЩЕН ЗНАКОМСТВУ С ДАТЧИКАМИ, УСТАНОВЛЕННЫМИ НА СПУТНИКЕ, ИХ ВЗАИМОДЕЙСТВИЮ МЕЖДУ СОБОЙ И С ИСПОЛНИТЕЛЬНЫМИ УСТРОЙСТВАМИ СПУТНИКА. ИСПОЛНИТЕЛЬНОЕ УСТРОЙСТВО – УСТРОЙСТВО СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ИЛИ РЕГУЛИРОВАНИЯ, ВОЗДЕЙСТВУЮЩЕЕ НА ПРОЦЕСС В СООТВЕТСТВИИ С ПОЛУЧАЕМОЙ КОМАНДНОЙ ИНФОРМАЦИЕЙ.

В ТЕОРЕТИЧЕСКОЙ ЧАСТИ РАССКАЗЫВАЕТСЯ О ФИЗИЧЕСКИХ ПРИНЦИПАХ РАБОТЫ ДАТЧИКОВ, ВИДАХ ДАТЧИКОВ И ФУНКЦИЯХ ДАТЧИКОВ, УСТАНОВЛЕННЫХ НА СПУТНИКЕ.

ТАКЖЕ РАССМАТРИВАЮТСЯ УСЛОВИЯ РАБОТЫ ДАТЧИКОВ, ВЗАИМОДЕЙСТВИЕ С ИСПОЛНИТЕЛЬНЫМИ УСТРОЙСТВАМИ СПУТНИКА И РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММ ВЗАИМОДЕЙСТВИЯ С ИСПОЛНИТЕЛЬНЫМИ УСТРОЙСТВАМИ СПУТНИКА.

В ПРАКТИЧЕСКОЙ ЧАСТИ ПОДРОБНО РАЗБИРАЮТСЯ ОСОБЕННОСТИ РАБОТЫ ДАТЧИКА ТОКА И ДАТЧИКА ИНТЕНСИВНОСТИ СВЕТА, А ТАКЖЕ ОСУЩЕСТВЛЯЕТСЯ СБОРКА СТЕНДА, ПРЕДСТАВЛЯЮЩЕГО СИСТЕМУ СБОРА ИНФОРМАЦИИ С ДАТЧИКОВ И ФОРМИРОВАНИЯ УПРАВЛЯЮЩЕГО СИГНАЛА ИСПОЛНИТЕЛЬНЫМ УСТРОЙСТВАМ. В КАЧЕСТВЕ ОСНОВНОЙ ФУНКЦИИ СИСТЕМЫ РАЗРАБАТЫВАЕТСЯ АЛГОРИТМ, АВТОМАТИЧЕСКИ ПОВОРАЧИВАЮЩИЙ СОЛНЕЧНУЮ БАТАРЕЮ К ИСТОЧНИКУ НАИБОЛЕЕ ЯРКОГО СВЕТА И РАССЧИТЫВАЮЩИЙ ПРИБЛИЗИТЕЛЬНОЕ ВРЕМЯ ЗАРЯДА БАТАРЕИ ОТ ДАННОГО ИСТОЧНИКА.



Практика I



РАБОТА В МАЛЫХ ГРУППАХ



4 ЧАСА

Разбираемся в принципе работы потенциометра

РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ.
Рекомендуется потратить 40 минут на рассказ об основных понятиях.

С одним набором может работать группа от двух до четырех человек. Разделите группу учащихся на подгруппы по 1-2 человека. Пусть одна подгруппа собирает электрическую цепь, а другая программирует плату управления. На это уйдет порядка 40 минут.

В конце занятия не лишним будет задать вопросы на закрепление материала и дать теоретическую справку для подготовки ко второму занятию (40 минут).

ВВЕДЕНИЕ И ТЕОРИЯ

Сейчас почти все устройства, окружающие нас, являются сложными системами, работающими частично или полностью независимо от человека. Автоматизация работы различных систем позволяет человеку взаимодействовать с устройствами, не углубляясь во все тонкости происходящего процесса. Однако каким образом возможна автоматическая работа таких систем? В данном модуле разбираются принципы работы с измерительными устройствами, являющимися неотъемлемой частью любой системы, имеющей даже малую степень автоматизации.

Для того, чтобы автоматические системы могли воспринимать информацию из внешнего мира, необходимы устройства, преобразующие разнообразные физические явления – например тепло, холода, давление и другие в электрические сигналы. Такими устройствами являются датчики. Иными словами, датчики – это глаза и уши технических систем.



3.1 СТРОЕНИЕ ДАТЧИКА.

Датчик – это интерфейсное устройство, воспринимающее внешние воздействия и реагирующее на них изменением электрических сигналов. Под внешним воздействием понимается количественная характеристика объекта, его свойство или качество, которые необходимо воспринять и преобразовать в электрический сигнал. Датчики в системах, созданных человеком, «разговаривают» с устройствами, с которыми они связаны, на одном языке. Язык общения – электрические сигналы, в которых информация передается при помощи электронов.

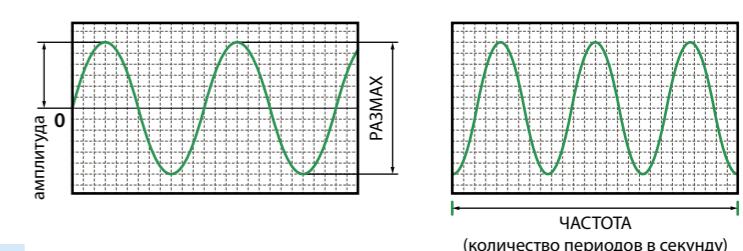
Датчик (sensor) состоит из двух частей: чувствительного элемента (detector) и преобразователя (transducer). Чувствительный элемент иногда называют измерительной головкой (рис. 3.1). Результат работы датчика – это реакция чувствительного элемента на внешнее воздействие, которая на выходе из преобразователя представляет собой электрический сигнал, распространяющийся далее по проводнику, (т.е. этот сигнал является выходным сигналом датчика).

Назначение датчиков – реакция на внешнее воздействие и преобразование его в электрический сигнал, совместимый с измерительными схемами. Другими словами, можно сказать, что датчик – это преобразователь физической величины в электрический сигнал.

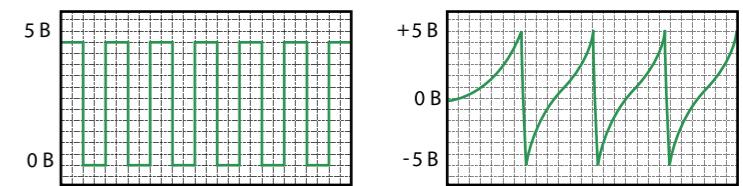
Сигналом называется изменение физической величины во времени. Часто сигналы изображают в виде графиков, отображающих изменение измеряемой величины.

Каждый сигнал задается несколькими характеристиками:

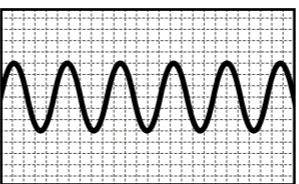
- амплитуда сигнала – максимальное значение сигнала.
- частота сигнала – количество повторений сигнала за секунду.



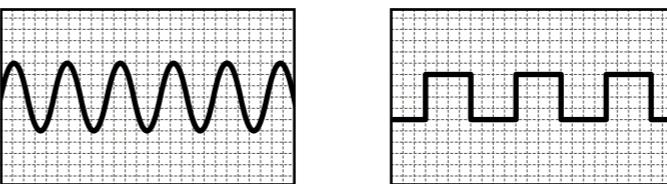
3.2



3.3



3.4



ЦИФРОВОЙ СИГНАЛ

Измеряемая датчиком величина пропорционально влияет на изменение выходной величины тока или напряжения (рис. 3.3).

Таким образом, каждый датчик характеризуется набором входных параметров любой физической природы и набором выходных электрических параметров*.

Столи также отметить, что сигналы бывают двух видов: аналоговые и цифровые.

Аналоговым является сигнал, непрерывно изменяющийся во времени и имеющий определенное значение в каждый бесконечно малый момент времени. По сути, вся информация, идущая от окружающего мира, является аналоговой: звук, свет и пр.

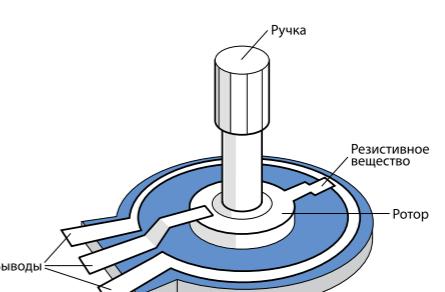
Цифровой сигнал имеет два значения – минимальное и максимальное. В случае работы с Arduino – это значение HIGH, соответствующее 5 В, и LOW, соответствующее 0 В. На рисунке 3.4. показан общий вид аналогового и цифрового сигналов.

Чтобы контролеры могли считывать аналоговый сигнал, используют аналого-

во-цифровые преобразователи, которые могут оцифровать сигнал. Тогда в каждый момент времени сигнал будет иметь цифровое значение, однако значения цифрового сигнала будут варьироваться в установленном диапазоне. По такому принципу строится работа с аналоговыми датчиками.

ПРОЕКТИРОВАНИЕ, ИЗГОТОВЛЕНИЕ

Рассмотрим работу датчиков на примере переменного резистора. Переменный резистор (потенциометр) – это резистор с регулируемым сопротивлением.



3.6

3.2 АМПЛИТУДА И ЧАСТОТА СИГНАЛА

3.3 ВЫХОДНЫЕ ХАРАКТЕРИСТИКИ ДАТЧИКА

3.4 ОБЩИЙ ВИД АНАЛОГОВОГО И ЦИФРОВОГО СИГНАЛОВ

3.5 ОЦИФРОВАННЫЙ СИГНАЛ

3.6 ПРИНЦИП ДЕЙСТВИЯ ПОТЕНЦИОМЕТРА

* Подробнее о параметрах датчиков можно почитать тут: <https://geektimes.ru/post/255116/>

Потенциометры используются как регуляторы различных параметров – громкости звука, мощности, напряжения и т.п. (рис. 3.6.)

При повороте ручки потенциометра происходит изменение сопротивления, то есть при этом создается деление тока. Напряжение изменяется вследствие увеличения или уменьшения величины сопротивления на определенном участке электрической цепи. Чем длиннее участок с резистивным веществом между выводами (контактами) резистора, тем выше сопротивление на этом участке.

РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ. Кроме теории работы переменного резистора (потенциометра), необходимо раздать его ученикам и предложить измерить его сопротивление с помощью мультиметра. Также можно разобрать переменный резистор для наглядной демонстрации его работы, показать анимированную и электрическую схемы и объяснить принцип работы электрической схемы.

Чтобы понять зависимость двух физических величин (силы тока от сопротивления), вспомним курс физики из школьной программы. Закон Ома для участка цепи: сила тока в участке цепи прямо пропорциональна напряжению на концах этого участка и обратно пропорциональна сопротивлению этого участка. Итак, мы вспомнили, что сила тока возрастает при падении сопротивления на определенном участке электрической цепи и наоборот. Это можно продемонстрировать на потенциометре.

В нашем случае сила тока будет оставаться неизменной, но при этом будет изменяться сопротивление резистора и, как следствие, – измеряемое напряжение на выходе.

МАТЕРИАЛЫ И ИНСТРУМЕНТЫ

Из инструментов понадобится мультиметр.

СОВЕТ: если вы не знакомы с программированием микроконтроллеров, можно предварительно посмотреть какие-либо вводные открытые уроки по Arduino, например, на <http://all-arduino.ru>. Для работы вам понадобится компьютер (ноутбук) с установленным на нем Arduino IDE (можно скачать бесплатно, например, с <http://arduino.cc>).

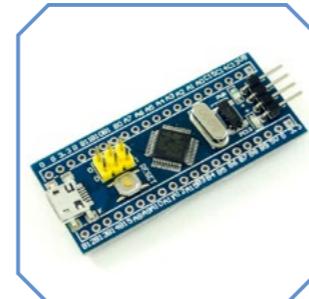
Хотя в наборе используется плата на базе STM32 Blue Pill, она является ардуино-совместимой, поэтому код, написанный для Arduino подходит и для неё.

ПОДГОТОВКА СРЕДЫ ПРОГРАММИРОВАНИЯ

В нашей работе мы будем использовать популярный инструмент Arduino IDE.



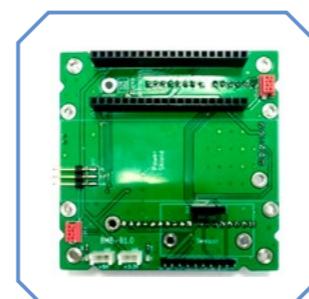
ПЕРЕМЕННЫЙ РЕЗИСТОР (ПОТЕНЦИОМЕТР), 1 ШТ.



ПЛАТА УПРАВЛЕНИЯ BLUEPILL, 1 ШТ.



ДАТЧИК ОСВЕЩЕННОСТИ, 1 ШТ.



МАТЕРИНСКАЯ ПЛАТА, 1 ШТ.



ПЛАТА ДЛЯ ПРОТОТИПИРОВАНИЯ, 1 ШТ.



ПРОГРАММАТОР ST-LINK, 1 ШТ.



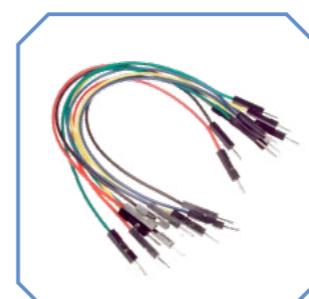
СВЕТОДИОД, 1 ШТ.



КОНВЕРТЕР USB-UART, 1 ШТ.



РЕЗИСТОР 220 ОМ, 1 ШТ.



СОЕДИНЯТЕЛЬНЫЕ ПРОВОДА

Однако, чтобы была возможность прошивать нашу плату с его помощью, его нужно подготовить.

1. На ПК помимо Arduino IDE следует поставить STM32CubeProgrammer. Использовать его напрямую мы не будем, однако сама Arduino IDE использует его элементы. Скачать и установить данное ПО можно по ссылке: <https://www.st.com/en/development-tools/stm32cubeprog.html>. ПО бесплатное, но чтобы его скачать, необходимо зарегистрироваться на сайте.

2. Саму Arduino IDE необходимо настроить следующим образом:

2.1. Перейти в меню **Файл -> Настройки**. Откроется окно как на рисунке 3.7. В строке «**Дополнительные ссылки для Менеджера плат**» нужно указать ссылку https://raw.githubusercontent.com/stm32duino/BoardManagerFiles/main/package_stmicroelectronics_index.json.

2.2. Далее перейти **Инструменты -> Плата -> Менеджер плат**. Откроется окно, как на рисунке 3.8. В списке необходимо найти и установить STM32 Cores.

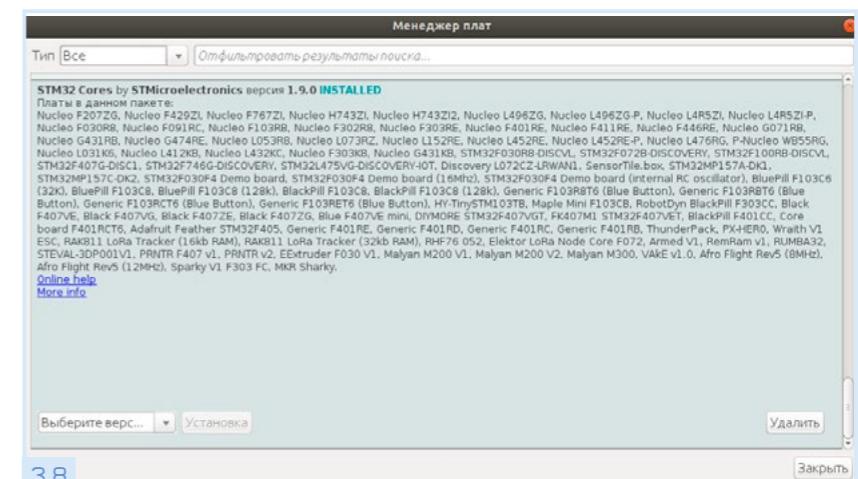
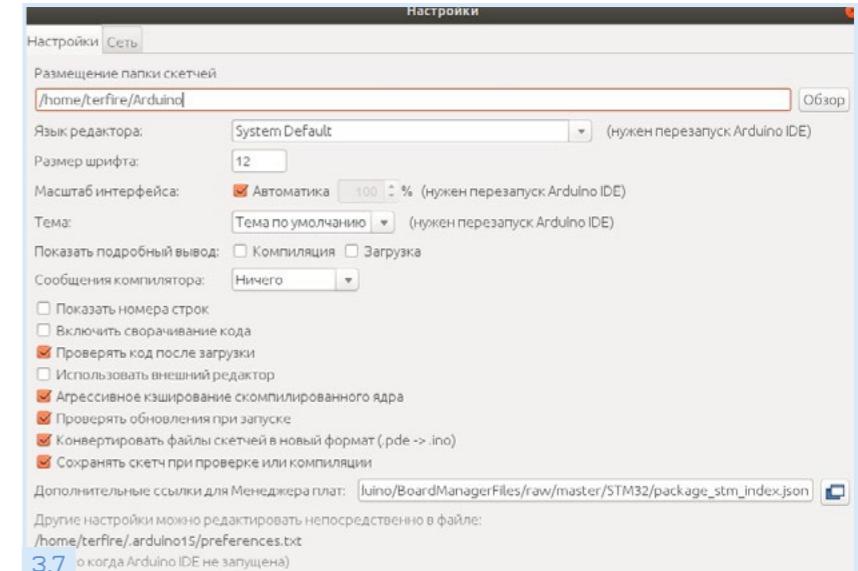
2.3. После в **Инструменты -> Плата** станут доступны платы STM32. Нас интересует «**Generic STM32 F1 series**». В **Board part number** нужно выбрать **BluePill F103C8**.

После этих действий станет возможна прошивка платы Blue Pill через Arduino IDE.

ПРОБНАЯ ПРОГРАММА

Любая программа, которая пишется для микроконтроллера, состоит из двух частей. Первая часть задается функцией «**setup**», и ее задача – задать перед началом работы все необходимые настройки. Функция «**setup**» выполняется микроконтроллером один раз. Вторая часть задается функцией «**loop**». После однократного выполнения функции «**setup**» микроконтроллер будет циклически выполнять функцию «**loop**» до тех пор, пока его не отключат от питания. В этой функции задается основная рабочая программа.

Самый простой код, который позволит проверить, работает ли прошивка устройства, – это мигание встроенным светодиодом на плате. Встроенный светодиод подключен к выходу микроконтроллера с называнием PC13. Следовательно, чтобы светодиод мигал, нужно чередовать два действия с определенным интервалом:



подачу питания и отключение питания.

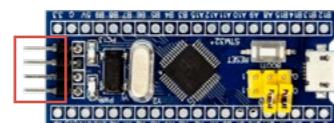
```
void setup() {
    //определяем pin PC13 как выходной
    pinMode(PC13, OUTPUT);
}
```

```
void loop() {
    //задаем низкий уровень на пине
    digitalWrite(PC13, LOW);
    //ожидаем 1 секунду
    delay(1000);
    //задаем высокий уровень на пине
    digitalWrite(PC13, HIGH);
    //ожидаем 1 секунду
    delay(1000);
}
```

Запишем полученную программу на плату управления Blue Pill. Для этого необходимо подсоединить к плате программатор, который позволяет прошивать плату. Программатор подсоединяется к Blue Pill по четырем проводам, выделенным в таблице.

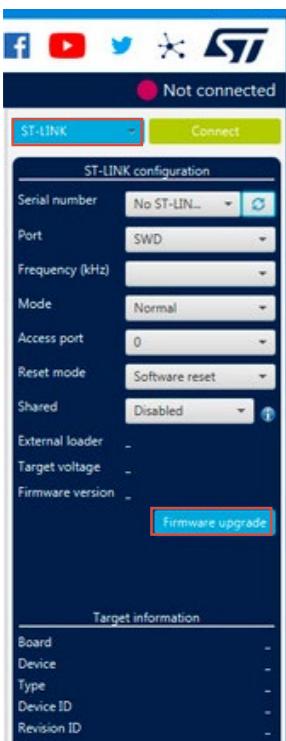
3.7 НАСТРОЙКИ МЕНЕДЖЕРА ПЛАТ В ARDUINO IDE

3.8 УСТАНОВКА STM32 CORES

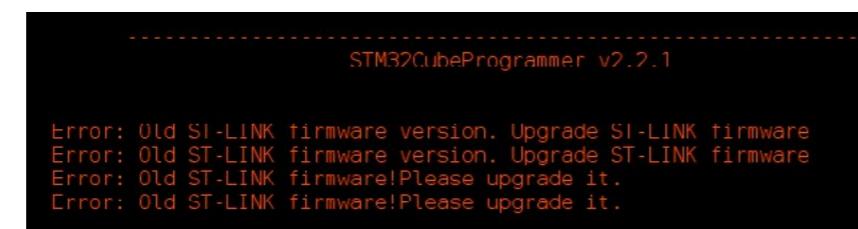
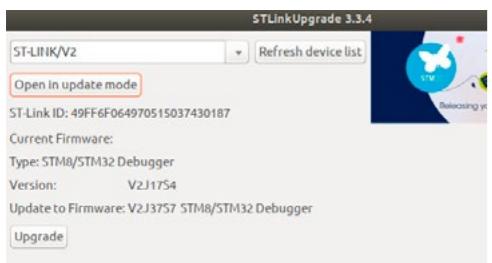
ST-Link V2	Blue Pill
	
(2) SWCLK > (4) SWDIO > (5) или (6) GND > (7) или (8) 3.3V >	→ SCK → DIO → GND → 3.3V

После подключения программатора становится возможна загрузка программы в плату. Возможно, ST-Link попросит обновления. Ошибка при попытке загрузки может выглядеть как показано на скриншоте справа.

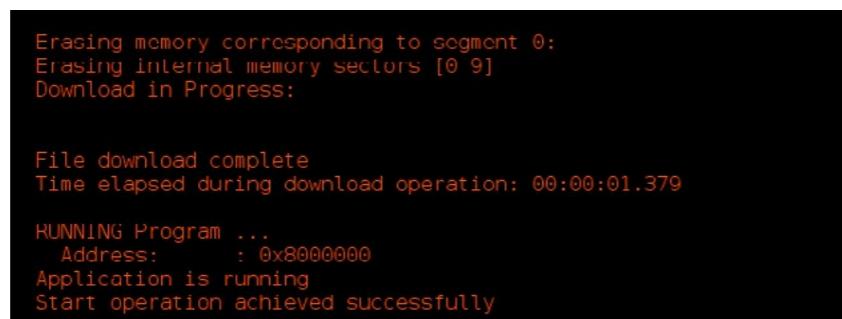
Для того, чтобы обновить ST-Link, запустите STM32CubeProgrammer. В правом меню в выпадающем списке выберите ST-Link и нажмите Firmware upgrade.



Появится новое окно.



Нажмите «Open in update mode», а затем «Upgrade». Вернитесь в Arduino IDE и снова попробуйте прошить плату. Должно появиться сообщение, наподобие:



После того как программа залита на плату – встроенный светодиод начнет мигать.

Если у вас возникли трудности на данном этапе, то рекомендуем вам посмотреть FAQ по ссылке [introsat.ru/faq](#). Если это не помогло, напишите нам в техподдержку по адресу info@introsat.ru.

НАСТРОЙКА UART

UART – широко используемый интерфейс, позволяющий передавать данные от одного устройства к другому. В нашем случае он нужен для того, чтобы считывать данные с датчиков, подключенных к плате BluePill и выводить их на экран ПК.

Более подробно ознакомиться с UART вы можете самостоятельно, например, изучив данный материал: <https://wiki.arduino.cc/page/serial-protocols-uart/>.

Чтобы воспользоваться UART на плате BluePill, необходимо подключить к плате конвертер usb-uart. Выходы микроконтроллера TX и RX расположены на обратной стороне материнской платы.

Их необходимо соединить с сигналами RXD (с Rx) и TXD (с Tx) конвертера соответственно. Оставшиеся два провода соединяются GND-GND и 3.3V-3.3V соответственно, для удобства, можно в том же ряду пинов. Перед подключением также проверьте напряжение на самом конвертере, где стоит перемычка: она должна соответствовать уровню используемого напряжения.

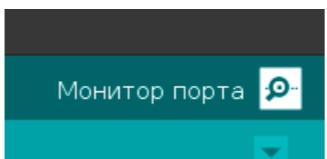
После подключения конвертера к плате необходимо также подключить его к usb-порту вашего ПК посредством USB кабеля.

Проверим передачу данных по UART. Для этого загрузим с помощью ST-Link новый код на плату:

```
void setup() {
    //Инициализируем работу UART
    Serial.begin(115200, SERIAL_8E1);
}

void loop() {
    //Отправляем строку текста в UART
    Serial.println("Hello World!");
    //ожидаем 1 секунду
    delay(1000);
}
```

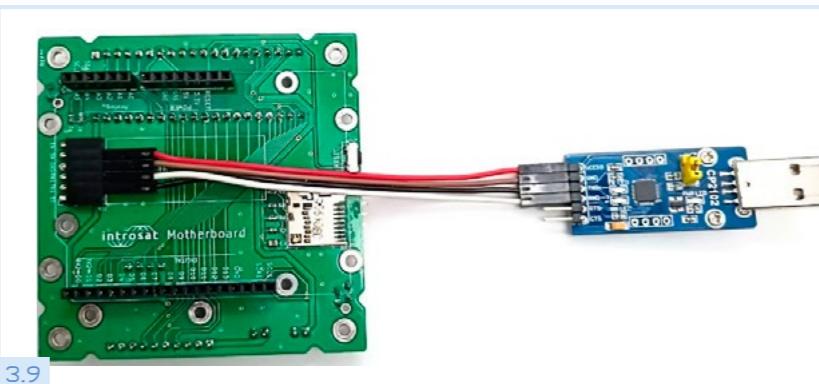
После загрузки программы нажмите на лупу в правом верхнем углу для того, чтобы открыть монитор порта.



Установите внизу открывшегося окна скорость 115200.



В мониторе порта должна отобразиться выводимая фраза.



3.9 ПОДКЛЮЧЕНИЕ UART

В приложении 3 приведена также инструкция по настройке беспроводного способа прошивки и чтения данных через Serial. Этот способ требует предварительной подготовки, однако, он гораздо более удобен непосредственно в работе.

Если у вас возникли трудности на данном этапе, то рекомендуем вам посмотреть FAQ по ссылке [introsat.ru/faq](#). Если это не помогло, напишите нам в техподдержку по адресу info@introsat.ru.

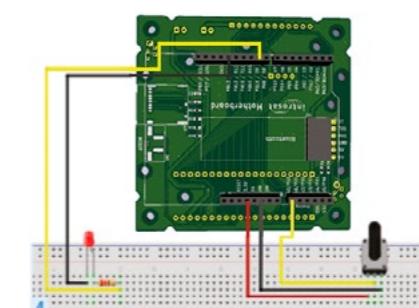
СБОРКА УСТРОЙСТВА

1. Соедините материнскую плату и плату BluePill, как показано на фото.

ВАЖНО! Разъем microusb на плате BluePill должен находиться над надписью «USB SIDE».



2. Соберите на макетной плате простейшую цепь, состоящую из светодиода, резистора и потенциометра для регулирования напряжения.



В нашей цепи от поворота ручки потенциометра будет зависеть яркость светодиода.

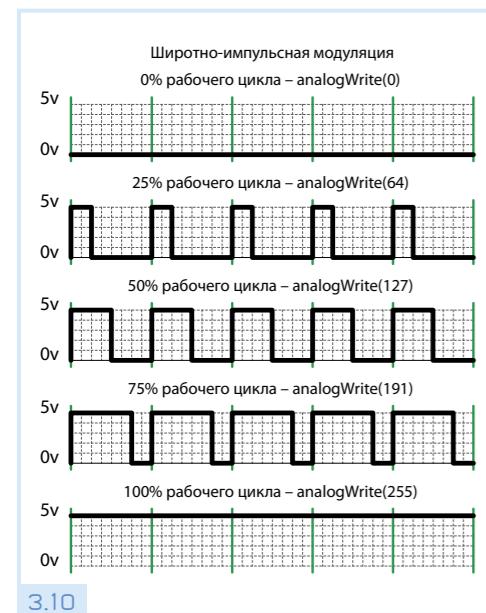
Напишем программу для правильной работы потенциометра. Программа решает три основные задачи:

- получение данных (значение напряжения) потенциометра;
- преобразование полученных данных в параметры для светодиода;
- передача результата на светодиод.

Для начала определяем «пины» (входы/выходы). «Пины», в соответствии с сигналами, бывают аналоговые (A0, A1, A2 и т. д.) и цифровые (D0, D1, D2, D3, D4 и т. д.).

ВАЖНО! На материнской плате подписаны также пины микроконтроллера STM32 рядом с наименованиями, соответствующими пинам Arduino. Для более корректной работы нужно указывать их в коде программы.

В нашем случае, потенциометр подключен к аналоговому сигналу, что логично, так как сопротивление будет изменяться плавно и непрерывно. Светодиод же подключен к цифровому с функцией ШИМ (широко-импульсной модуляции). ШИМ позволяет с помощью цифрового сигнала варьировать значение с помощью генерируемых импульсов. Импульс, как и цифровой сигнал, может принимать всего два значения: 0 или 3.3V. У каждого импульса есть длительность, а у череды импульсов есть частота, с которой они генерируются. Меняя длительность импульса и частоту импульсов, можно получить среднее значение между двумя уровнями 0 и 3.3V.



3.10

Управление уровнем ШИМ задается в диапазоне 0-255, где значение 255 соответствует подаче постоянного напряжения в 3.3V на всем промежутке времени (рис. 3.9) и будет соответствовать максимальной светимости светодиода. Если значение меньше 255, то яркость светодиода задается чередой импульсов.

Пример программы:

```
//даем имена пинов со светодиодом
#define led PB11
//и потенциометром
#define pot PA0
```

Установим режим работы заданных «пинов» внутри функции setup():

```
void setup(){
    //put your setup code here, to run once:
    //pin со светодиодом – выход
    pinMode(led, OUTPUT);

    //pin с потенциометром – вход
    pinMode(pot, INPUT);
}
```

При этом «пин», к которому подключен потенциометр, будет работать в режиме приема значения (плата принимает значения от потенциометра), а «пин» светодиода – в режиме отправки значения (плата управления передает значения светодиоду).

Далее определим переменную *x*, которая будет хранить значения потенциометра и переменную *y*, которая будет определяться как значение для светодиода.

```
void loop()
{
    int x;
    int y;
    // считываем напряжение с потенциометра:
    // будет получено число
    // от 0 до 1023
    x = analogRead(pot);
    // делим его на 4, получится число в
    // диапазоне 0-255
    // (дробная часть будет отброшена)
    y = x / 4;
    // выдаём результат на светодиод:
    digitalWrite(led, y);
}
```

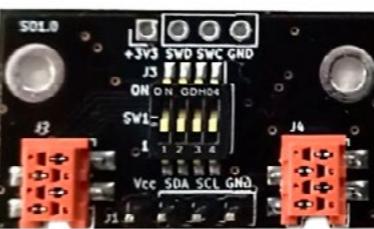
Проверим работу программы, повернув потенциометр сначала в одну сторону, а потом в другую. Если все правильно, яркость светодиода будет меняться.

3.10 ШИМ

* Более подробно про ШИМ:
<http://arduino.ru/Tutorial/PWM>

ИЗМЕРЕНИЕ ОСВЕЩЕННОСТИ

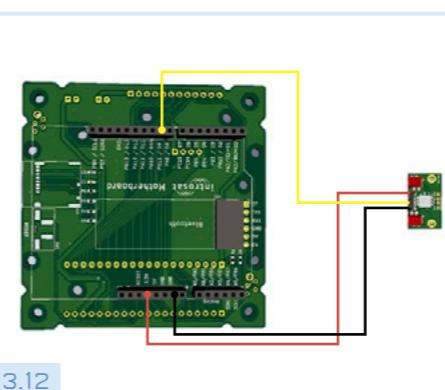
Датчик освещенности позволяет измерять силу светового потока, который падает на него. Датчик работает в нескольких режимах. Для начала разберем первый режим работы. Для этого датчик необходимо подключить к плате Blue Pill по трем проводам: Vcc, SCL и GND.



3.11

Выходным результатом работы датчика является сигнал ШИМ; чем больше значение силы светового потока, падающего на датчик, тем короче конфигурируемый импульс. Длительность импульсов является выходной информацией для датчика, и ее значения варьируются в диапазоне 0-1800 [мс].

Перед подключением датчика к плате убедитесь, что все переключатели стоят так, как показано на фото (т.е. в нижнем положении). Такая конфигурация датчика соответствует определенному режиму работы. Для подключения датчика освещенности используем соединительные провода «мама-папа». Провода должны быть подключены следующим образом (рис. 3.12):



3.12

- контакт «SCL» подключается к цифровому «pin» (например D9);
- контакт «Vcc» подключается к контакту «3.3V»;
- контакт «GND» подключается к контакту «GND».

Далее подключим плату Blue Pill к ком-

пьютеру и напишем программу для получения сигнала от датчика освещенности.

Определим «pin» датчика и переменную, в которую будем записывать, получаемую с него информацию.

```
#define light_sensor1 PB11
```

Далее внутри функции setup установим режим работы заданного вход/выхода (pin) как входа.

```
pinMode(light_sensor1, INPUT);
```

Зададим настройки ком-порта:

```
Serial.begin(115200, SERIAL_8E1);
```

Чтобы получить и отобразить информацию, передаваемую датчиком, внутри функции loop, напишем следующий код:

```
//записать сигнал в переменную light1
long light1 = pulseIn(light_sensor1, HIGH);
```

```
//отправить переменную в ком-порт для
//отображения на ПК
Serial.println(light1);
```

Далее необходимо подключить к плате конвертер usb-uart, как описывалось ранее.

Проверим программу и запишем ее в плату. Откроем монитор порта для отображения результата работы программы. Результатом работы программы является вывод числовых значений от 0 до 1800, где 0 – наиболее интенсивное освещение, а 1800 – наименее интенсивное.

РЕЗУЛЬТАТ

В результате можно увидеть, как изменяется яркость светодиода при повороте потенциометра. Таким образом, получилась простейшая система управления, с принципами работы которой мы более подробно познакомимся на следующем занятии.

Помимо этого учащиеся знакомятся со считыванием данных с датчиков освещенности.

ПОЛУЧАЕМЫЕ НАВЫКИ

Учащиеся осваивают физические принципы работы датчиков, получают навыки программирования датчиков на примере переменного резистора и датчика освещенности.

3.11 ДАТЧИК ОСВЕЩЕННОСТИ

3.12 СХЕМА ПОДКЛЮЧЕНИЯ ДАТЧИКА ОСВЕЩЕННОСТИ

Знания к практике I (см. ниже) являются основной теоретической информацией для практической части и должны быть представлены и записаны учениками во время занятия.

На следующем занятии рекомендуется проверить знание материала. График (рис. 3.13) с пояснениями рекомендуется изобразить во время представления данной теории.

Знания к практике I



ТЕОРИЯ УПРАВЛЕНИЯ

Для каждого датчика можно вывести идеальное или теоретическое математическое соотношение, связывающее величины сигналов на входе и выходе. Если бы была возможность спроектировать и изготовить датчик с математической точностью, без погрешности, то сигнал на выходе такого датчика

всегда бы соответствовал реальному значению внешнего воздействия. Выведенное идеальное соотношение между входным и выходным сигналом можно выразить в виде таблицы, графика (рис. 5) или математического выражения. Это идеальное (теоретическое) выражение часто называют передаточной функцией.

Передаточная функция устанавливает взаимосвязь между выходным электрическим сигналом датчика S и внешним воздействием x датчика:

$$S = f(x).$$

Передаточная функция может быть линейной и нелинейной.

Линейная передаточная функция

P. S.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru



3.13 ПРИМЕР ЛИНЕЙНОЙ ПЕРЕДАТОЧНОЙ ФУНКЦИИ

отображает прямую зависимость электрического сигнала от внешнего воздействия, вычисляемую по формуле:

$$y=kx+b, \text{ где } k \text{ и } b \text{ – числа.}$$

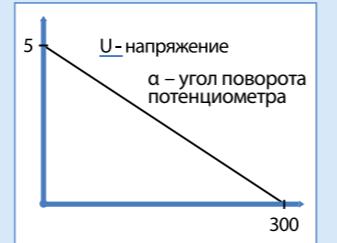
Нелинейная передаточная функция может быть, к примеру, степенной или логарифмической. Такие функции зачастую описаны в документации к датчикам.

РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ

Привести в качестве примера передаточную функцию переменного резистора (потенциометра).

Потенциометр, используемый в данном занятии, имеет линейную передаточную функцию, отражающую

зависимость угла поворота ручки потенциометра. В зависимости от угла поворота ручки, сопротивление потенциометра увеличивается, а сила тока остается неизменной, следовательно, выходное напряжение уменьшается.



ПРИМЕР ГРАФИКА ЛИНЕЙНОЙ ПЕРЕДАТОЧНОЙ ФУНКЦИИ

Практика II



РАБОТА В МАЛЫХ ГРУППАХ



2 ЧАСА

Принцип работы датчика освещенности. Сборка стенда солнечной батареи.

СОВЕТЫ ПРЕПОДАВАТЕЛЮ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ: Уделите 15 минут тому, чтобы вспомнить понятие передаточной функции (см. «Знания к практике I»). Далее следует отвести 30 минут на рассказ о принципах работы датчика освещенности. Также можно рассказать и о принципах работы температурных датчиков, датчиков движения и давления. Практическое занятие занимает около 60 минут. С одним набором компонент могут работать от 2 до 4 человек. Группы также удобно разделить на подгруппы по 1-2 человека. Одни подгруппы могут собирать корпус, другие – электрическую схему. В конце занятия потратите 15 минут на закрепление материала. Начиная с этого раздела и далее приводятся примеры кодов для работы с использованием bluetooth-модуля. Если вы пропустили настройку bluetooth, то посмотрите описание в приложении 3.

ВВЕДЕНИЕ И ТЕОРИЯ ВНУТРЕННИЙ ФОТОЭФФЕКТ

Датчик интенсивности света используется для измерения интенсивности света, падающего на чувствительный элемент. В основе его работы лежит эффект фотопроводимости.

Фотопроводимость – явление изменения электропроводности вещества при поглощении электромагнитного излучения (видимое, инфракрасное, ультрафиолетовое или рентгеновское излучение).

При поглощении света возникают избыточные носители заряда, приводящие к увеличению проводимости полупроводника. Процесс внутреннего освобождения электронов под действием света называется внутренним фотоэффектом.

На основе внутреннего фотоэффекта основаны измерительные устройства, такие как фотодиоды, фототранзисторы и фотодиоды. В основе каждого из них лежит полупроводник.

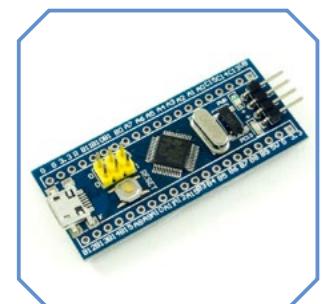
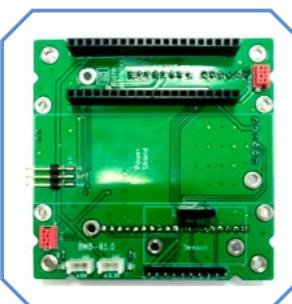
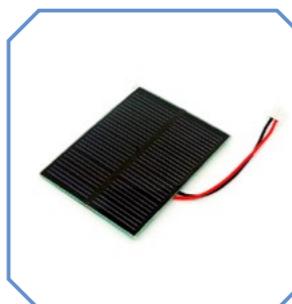
Проводимость, приобретаемая полупроводником при облучении светом, называется фотопроводимостью. На основе этого эффекта созданы фотодиоды. Чувствительный элемент фотодиода представляет собой пленку монокристаллического (цельный кристалл с единой

кристаллической решеткой) или поликристаллического (множество кристаллов) полупроводника с двумя контактами, нанесенными на основание, не проводящее электрический ток. Под действием напряжения, приложенного к фоточувствительному полупроводниковому элементу, созданные светом носители заряда дрейфуют и создают в электрической цепи ток, который называют фототоком I_ϕ . После прекращения облучения проводимость полупроводника за промежуток времени, равный времени жизни носителей t , возвращается к тому значению, которое имела до облучения.

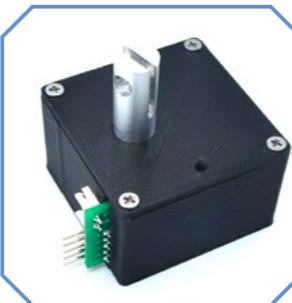
Другим полупроводниковым элементом является фотодиод. При воздействии света на фоточувствительный элемент, также как и в фоторезисторе, возникает фототок. Однако в отличие от фоторезистора, освещенность измеряется не по изменению проводимости, а по силе возникающей ЭДС.

Фототранзисторы отличаются от фотодиодов тем, что могут усиливать протекающий через них электрический ток пропорционально возникающему фототоку, что делает их гораздо более чувствительными по сравнению с фотодиодами.

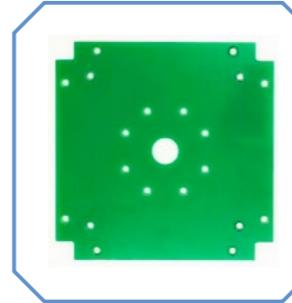
Датчик освещенности, с которым мы работали на прошлом занятии построен как раз на фототранзисторе.

ПЛАТА УПРАВЛЕНИЯ
BLUEPILL, 1 ШТ.МАТЕРИНСКАЯ
ПЛАТА, 1 ШТ.СОЛНЕЧНАЯ
БАТАРЕЯ, 1 ШТ.

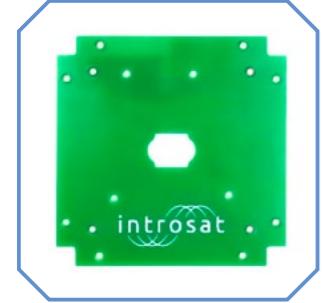
МОДУЛЬ ПИТАНИЯ, 1 ШТ.

АККУМУЛЯТОР 18650,
1 ШТ.ШАГОВЫЙ ДВИГАТЕЛЬ
В КОРПУСЕ, 1 ШТ.ЦИФРОВОЙ
АМПЕРМЕТР, 1 ШТ.ДАТЧИКИ
ОСВЕЩЕННОСТИ, 4 ШТ.ПЕРЕКЛАДИНА
С ПАЗОМ, 4 ШТ.ПЕРЕКЛАДИНА С ТОРЦЕВЫМ
ОТВЕРСТИЕМ, 4 ШТ.

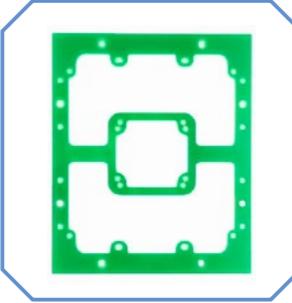
РЕЛЬС, 4 ШТ.



НИЖНЯЯ ПАНЕЛЬ, 1 ШТ.



ВЕРХНЯЯ ПАНЕЛЬ, 1 ШТ.



БОКОВАЯ ГРANЬ, 4 ШТ.



НАБОР КРЕПЕЖА M2.5

НАБОР СТОЕК ДЛЯ
ПЕЧАТНЫХ ПЛАТ M2.5

ПРОВОД ПИТАНИЯ, 1 ШТ.

НАБОР МОНТАЖНЫХ
ПРОВОДОВ

I.a



I.b



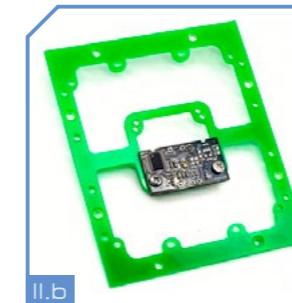
I.c



I.d



II.a



II.b



II.c



II.d

СОВЕТ: Спросите у учеников, зачем спутникам могут понадобиться датчики освещенности. Правильным ответом будет утверждение, что по датчикам, реагирующим на свет, можно определить положение Солнца. А зная положение Солнца, можно развернуть к нему солнечные батареи, чтобы с их помощью получать энергию, необходимую для автономной работы спутника на орбите.

СБОРКА УСТРОЙСТВА

I. Сборка рамы спутника

I.a Возьмите два рельса и установите в выемки перекладины с пазом.

I.b В получившиеся углубления на стыке рельса и перекладины установите перекладины с торцевым отверстием так, чтобы поперечные отверстия в этих перекладинах были ориентированы перпендикулярно рельсу (они используются для установки боковых панелей).

I.c Закрепите угловое соединение винтами M2.5x12 с потайной головкой.

I.d Установите две оставшиеся перекладины и два оставшихся рельса аналогичным образом и окончательно соберите конструкцию при помощи винтов M2.5x12.

II. Сборка материнской платы и боковых панелей

II.a Прикрепите шаговый двигатель к ниж-

ней панели, используя два винта M2.5x12 с потайной головкой. Вал двигателя при этом должен «выглядывать» через отверстие в центре панели. Для удобства дальнейшей коммутации можно (но не обязательно) приклеить одну из макетных плат при помощи двустороннего скотча, наклеенного на обратную сторону.

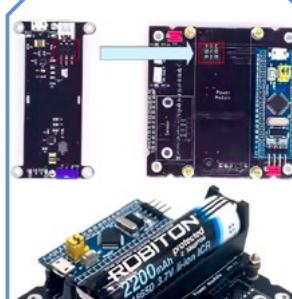


II.e

II.b Установите датчики освещенности на боковые панели, используя винты M2.5x6 с полукруглой головкой и гайки.

II.c Установите плату Blue Pill на материнскую плату.

ВАЖНО: Разъем microusb на плате BluePill должен находиться над надписью «USB SIDE».

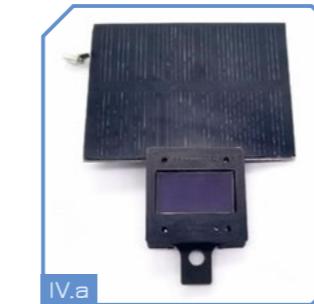
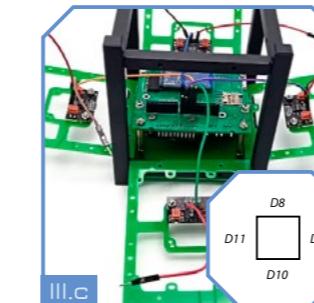
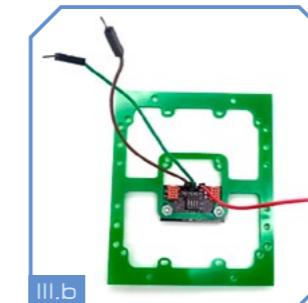
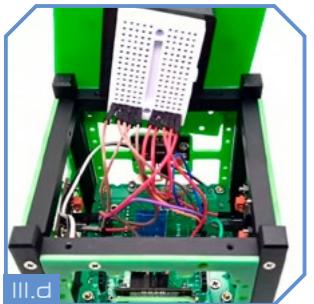
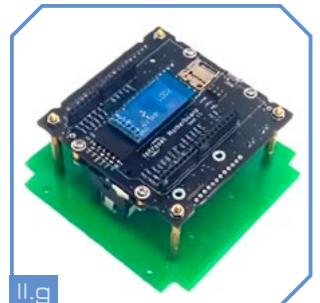


II.f

II.d Установите модуль Bluetooth в соответствующий разъем (если планируете использовать беспроводное подключение).

II.e Закрепите 4 стойки PCHSN5 (длиной 5 мм) на материнской плате при помощи винтов M2.5x4 с полукруглой головкой.

II.f Установите аккумулятор в модуль питания, при необходимости закрепите его пластиковой стяжкой через ушки в плате. Модуль питания установите на материнскую плату так, чтобы шестиконтактный разъем сопрягались друг с другом.



Закрепите гайками модуль питания.

II.9 Установите собранную материнскую плату на верхнюю панель с помощью межплатных стоек PCHSN-11 (для этого нужно будет соединить стойки между собой по 3 шт.) и гаек.

III. Окончательная сборка и подключение

III.a Закрепите верхнюю панель на раме с помощью винтов M2.5x6 с потайной головкой.

III.b К каждому датчику подсоедините три провода к контактам GND, SCL и Vcc, как описывалось на предыдущем занятии.

III.c Каждый контакт SCL с датчика подсоедините к цифровому pinу, например, D8, D9, D10, D11. Установите соответствие между гранью и pinом, чтобы понимать расположение датчиков на кубике.

III.d Закрепите боковые панели на раме с помощью винтов M2.5x6 с потайной головкой. Все контакты датчиков GND подсоедините к линейке pinов GND на платеprotотипирования, а pinы Vcc – к 3,3V.

III.e Входы шагового двигателя I4, I3, I2, I1 подключите к pinам D3, D4, D5 и D6 материнской платы. Выход SW1 соедините с pinом D7. Подключите выход питания 5В материнской платы к разъему питания шагового двигателя при помощи соответствующего кабеля.

III.f Закрепите нижнюю панель на раме винтами M2.5x6.

IV. Подготовка поворотной конструкции с солнечной батареей

IV.a Вставьте солнечную батарею в паз на торце корпуса амперметра. Присоедините провода от солнечной батареи к свободным контактам с обратной стороны амперметра, «+» к «+», «-» – к «-». Для включения переведите переключатель в положение «ON».



IV.b

IV.b Поставьте собранный корпус на стол так, чтобы ось шагового двигателя смотрела вверх и установите конструкцию из амперметра с солнечной батареей на ось.

Стенд солнечной батареи представляет собой систему управления, способную определить самый сильный источник освещения, повернуть батарею к этому источнику и определить время до полного заряда. Определение источника освещения осуществляется путем измерения датчиками интенсивности света. Поворот солнечной батареи осуществляется посредством передачи управляющего сигнала двигателю, осуществляющему этот поворот.

РЕЗУЛЬТАТ

В результате получаем готовый стенд солнечной батареи. Чтобы стенд начал работать, необходимо запрограммировать плату управления.

ПОЛУЧАЕМЫЕ НАВЫКИ

Учащийся осваивает навык аккуратной сборки и развивает пространственное мышление.

РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ

В качестве домашнего задания можно предложить ученикам продумать алгоритм работы стендов. На следующем заня-

тии можно рассмотреть предложенные учениками методы и реализовать их для проверки работоспособности.



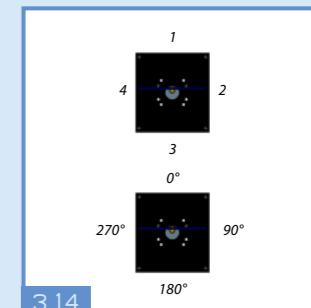
ПРОГРАММИРОВАНИЕ:
АЛГОРИТМИЗАЦИЯ

Важным условием работы стендов солнечной батареи будет проектирование оптимального алгоритма работы системы датчиков. Система состоит из четырех датчиков, расположенных на сторонах куба. Предполагая, что источник освещения направлен перпендикулярно к стороне, а значит, и расположенному на ней датчику, можно создать систему управления, проверяющую состояние каждого датчика через некоторый промежуток времени и поворачивающую солнечную батарею параллельно наиболее освещенному датчику.

В результате у солнечной батареи будет только три состояния:

- поворот на 0° (параллельно датчику 1);
- поворот на 90° по часовой стрелке (параллельно датчику 2);
- поворот на 180° по часовой стрелке (параллельно датчику 3);
- поворот на 270° по часовой стрелке (параллельно датчику 4)

3.14 СХЕМА ВРАЩЕНИЯ
СТЕНДА СОЛНЕЧНОЙ БАТАРЕИ



Практика III



РАБОТА В МАЛЫХ
ГРУППАХ



4 ЧАСА

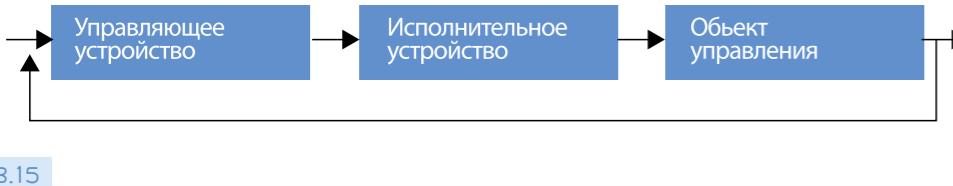
Программирование стенда умной солнечной батареи

РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ: Перед началом занятия рекомендуется вспомнить устройство собранного стенда. Далее отведите время на рассказ о принципе работы шагового двигателя, а затем на практике изучите принципы управления им. Работать с кодом ученикам лучше в парах за одним ПК. Проверять скетчи можно попарно на одном собранном стенде. Не лишним будет совместно с учениками нарисовать блок-схему алгоритма и только потом перейти к разбору кода. В конце стоит дать небольшое самостоятельное задание по включению в скетч кода для датчика тока, опорой в этом вам послужит занятие по практике II.

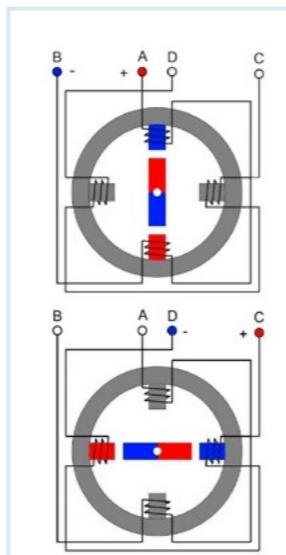
ВВЕДЕНИЕ И ТЕОРИЯ

Когда стоит задача не просто снять данные об окружающей среде, но и отреагировать на изменение этих данных, в системе измерения появляется еще один вид элементов – исполнительные устройства. Исполнительное устройство – устройство системы автоматического управления (или регулирования), воздействующее на происходящий в системе процесс в соответствии с получаемыми командами. Состоит из двух функциональных блоков: собственно, исполнительного устройства (если исполнительное устройство механическое, его часто называют «исполнительным механизмом») и регулирующего органа (рис. 3.15).

Под исполнительным устройством в теории понимают устройство, передающее



3.16 УСТРОЙСТВО
И ПРИНЦИП РАБОТЫ
ШАГОВОГО ДВИГАТЕЛЯ



3.16

Существует несколько режимов управления двигателем. Рассмотрим режим, в котором ток подается на две соседние обмотки (см. рисунок 3.17).

Наш стенд солнечной батареи содержит шаговый двигатель в качестве исполнительного устройства, который должен осуществлять поворот солнечной батареи к наиболее яркому источнику освещения. Расположение наиболее яркого источника освещения определяется путем сравнения уровня освещенности, определяемого датчиками, установленными на стенде (направленными в разные стороны под углом 90° друг относительно друга).

МАТЕРИАЛЫ И ИНСТРУМЕНТЫ

Собранный стенд солнечной батареи, включающий плату управления, четыре датчика интенсивности света, датчик тока, шаговый двигатель.

ПРОЕКТИРОВАНИЕ, ИЗГОТОВЛЕНИЕ

Напишем программу поворота солнечной батареи параллельно наиболее освещенному датчику. Для этого нужно дополнить написанную ранее программу.

Добавим переменные для управления шаговым двигателем.

```
#define in1 PB5
#define in2 PB4
#define in3 PB3
#define in4 PA15
```

Подключим библиотеку для использования bluetooth-модуля.

```
#include <IS_Bluetooth.h>
```

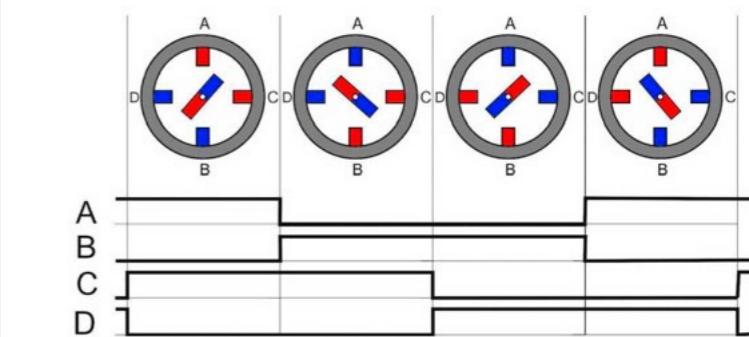
Библиотеку «IS_Bluetooth» и другие библиотеки можно скачать по ссылке: introsat.ru/download

Далее внутри функции *setup* зададим начальные настройки.

```
Serial.begin(115200, SERIAL_8E1);
```

В этой же функции установим режим работы входов/выходов (*pin*) как входов для датчиков освещенности.

```
pinMode(PA8, INPUT);
pinMode(PB11, INPUT);
pinMode(PB10, INPUT);
pinMode(PA11, INPUT);
```



А также пины для управления шаговым двигателем.

```
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
```

Определим еще один дополнительный pin для определения начального положения шагового двигателя.

```
pinMode(PC15, INPUT);
```

Добавим запись полученных значений в переменные и вывод значений в ком-порт в функцию *loop()*.

```
//код для прошивки через bluetooth-модуль
if(Serial.available()){
    char command = Serial.read();
    if(command == 'b'){
        enter_bootloader();
    }
}

long light1 = pulseIn(PA8, HIGH);
Serial.print(light1);
Serial.print("\t");
long light2 = pulseIn(PB11, HIGH);
Serial.print(light2);
Serial.print("\t");
long light3 = pulseIn(PB10, HIGH);
Serial.print(light3);
Serial.print("\t");
long light4 = pulseIn(PA11, HIGH);
Serial.println(light4);
```

Теперь мы можем увидеть полученные значения, открыв монитор порта.

Напишем программу, которая позволит управлять шаговым двигателем. Каждый из четырех проводов будет управлять обмоткой внутри двигателя. Напишем перед функцией *loop()* вспомогательный

код, который будет осуществлять включение и выключение обмоток, тем самым обеспечивая поворот вала двигателя:

```
//функция, задающая поворот на
//заданный угол против часовой стрелки
void left_rotation(int alpha)
{
    for (int i = 1; i <= ceil(alpha * 52 / 36); i++)
    {
        //подача напряжения на 3 и 4 pin
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, HIGH);
        delay(10);

        //подача напряжения на 4 и 1 pin, чтобы
        //осуществить проворот
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, HIGH);
        delay(10);

        //подача напряжения на 1 и 2 pin, чтобы
        //осуществить проворот
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, HIGH);
        delay(10);

        //подача напряжения на 3 и 4 pin, чтобы
        //осуществить проворот
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, HIGH);
        delay(10);

        //подача напряжения на 2 и 3 pin,
        //полный цикл завершится, если снова
        //подать напряжение на 3 и 4 pin
    }
}
```

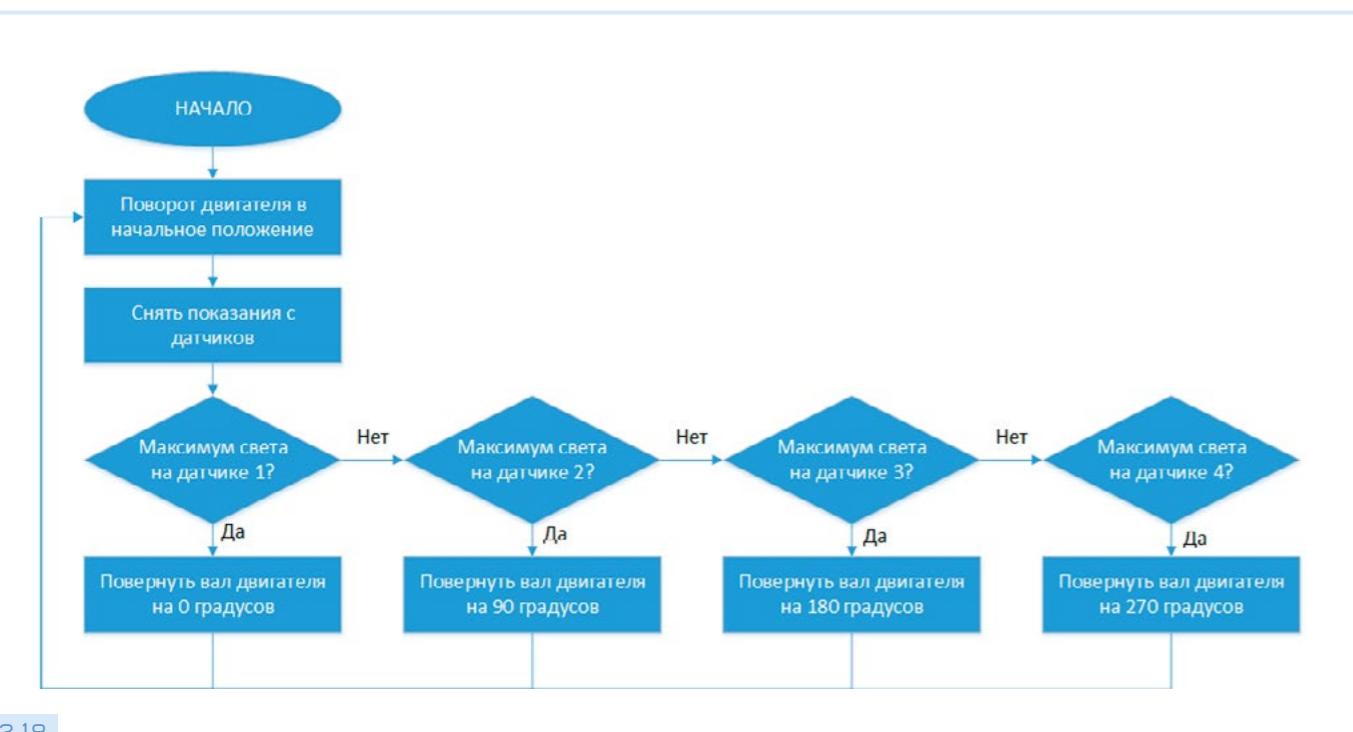
```
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(10);
}

//функция, задающая поворот на
//заданный угол по часовой стрелке
void right_rotation(int alpha)
{
    for (int i = 1; i <= ceil(alpha * 52 / 36); i++)
    {
        //подача напряжения на 1 и 2 pin
        digitalWrite(in1, HIGH);
        digitalWrite(in2, HIGH);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
        delay(10);

        //подача напряжения на 1 и 4 pin, чтобы
        //осуществить проворот
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, HIGH);
        delay(10);

        //подача напряжения на 3 и 4 pin, чтобы
        //осуществить проворот
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, HIGH);
        delay(10);
    }
}
```

3.18 БЛОК-СХЕМА РАБОТЫ СТЕНДА С СОЛНЕЧНОЙ БАТАРЕЕЙ



3.18

// подача напряжения на 2 и 3 pin,
// полный цикл завершится, если снова
// подать напряжение на 1 и 2 pin

```
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(10);
}

// снимаем показания с датчиков освещенности
long light1 = pulseIn(PA8, HIGH);
Serial.print(light1);
Serial.print("\t");
long light2 = pulseIn(PB11, HIGH);
Serial.print(light2);
Serial.print("\t");
long light3 = pulseIn(PB10, HIGH);
Serial.print(light3);
Serial.print("\t");
long light4 = pulseIn(PA11, HIGH);
Serial.println(light4);
```

Далее определим условие вращения двигателя внутри функции *loop*. Для удобства и наглядности составим блок-схему алгоритма разворота. Исходя из того, что датчики закреплены на гранях в том же порядке, какой указан в предыдущем занятии, можно сказать, у солнечной батареи четыре состояния:

- повернута на датчик 1
- повернута на датчик 2
- повернута на датчик 3
- повернута на датчик 4

Задача заключается в том, чтобы развернуть солнечную батарею к максимально освещенной стороне. Это может быть сторона, на которой закреплен датчик 1, датчик 2, датчик 3 или датчик 4.

Имея информацию о начальном положении солнечной батареи и о том, куда ее нужно повернуть, можно расписать условие (см. рис. 3.18).

Код для функции *loop()*, соответствующий блок-схеме, приведен ниже.

```
void loop()
{
    // код для прошивки через bluetooth-модуль
    if(Serial.available())
    {
        char command = Serial.read();
        if(command == 'b'){
            enter_bootloader();
        }
    }

    // поворот двигателя в начальное
    // положение //
    bool cont = digitalRead(D7);

    while(cont)
    {
        left_rotation(5);
        cont = digitalRead(D7);
    }

    // конец поворота двигателя в начальное
    // положение //
}
```

Из-за конструктивных особенностей шагового двигателя, возможно, понадобится некоторая поправка поворота двигателя на 10-45°. Для того, чтобы скорректировать эту поправку, необходимо добавить ее к повороту на определенный угол, т.е., если при сборке стендса получается, что солнечная батарея встает под углом в 15° к грани, нужно поворачивать батарею не на 0°, 90°, 180° и 270°, а на 15°, 105°, 195° и 285°. Необходимую корректировку легко внести в алгоритм.

Чтобы получить на солнечной батарее ток, значение которого отобразится на экране амперметра, необходимо установить источник света с лампой накаливания на расстоянии 10 см от грани собранного куба. Чтобы проверить работу алгоритма,

можно несколько раз изменить положение источника света. Каждый раз, при смене положения источника света, солнечная батарея должна стремиться повернуться к нему.

§ Возможно, ученикам потребуется время, чтобы правильно установить положение солнечной батареи относительно датчиков, так как у шагового двигателя есть свое нулевое положение, относительно которого тот начинает отсчет своего вращения.

РЕЗУЛЬТАТ

Результатом является программа поворота солнечной батареи к наиболее яркому источнику освещения.

РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ

Подводя итоги, следует рассказать ученикам о методах ориентации спутника по солнцу, а также пояснить, что системы, подобные собранной, установлены на спутниках, но имеют более сложный алгоритм работы.

ПОЛУЧАЕМЫЕ НАВЫКИ

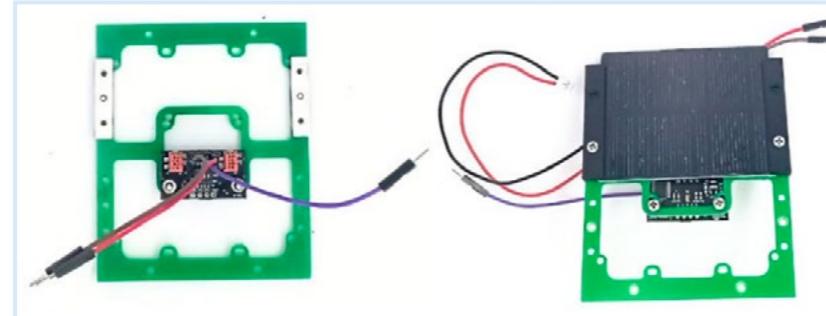
Ученики знакомятся с принципом работы шагового двигателя, а также разрабатывают программу управления исполняющим устройством на основе системы датчиков.

ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

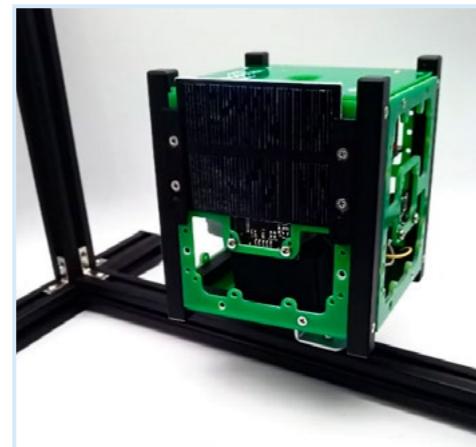
В качестве дополнительного задания можно усовершенствовать алгоритм управления, введя в него отсчет угла поворота. Это позволит не возвращаться каждый раз в нулевое положение, а просто доворачиваться на нужный угол.

При написании своего алгоритма учите, что поворачиваться от нулевого положения влево двигатель не может, так как там стоит ограничитель поворота. Поэтому отсчитывайте угол от нулевого положения и вращайте двигатель вправо. Если же необходимо повернуть двигатель влево, то учитывайте, что задаваемый угол поворота не должен превышать текущий.

Помимо этого, можно установить солнечную батарею не в качестве поворотной конструкции, а закрепить её на одной из граней с помощью соответствующих креплений (см. раздел «Ориентация в космическом пространстве», стр. 64-72).



Сам собранный кубсат прикрепить к нижней части подвеса. Сборка подвеса приведена в приложении 2.



В таком случае, чтобы поворачивать солнечную батарею к свету, необходимо разворачивать весь кубсат целиком.

СОВЕТ: Проведите небольшой эксперимент: пересоберите установку и запустите уже написанный код управления. Будет программа работать корректно и почему?

ЧТО ДАЛЬШЕ?

В текущем разделе мы начали знакомиться с устройством космического аппарата и алгоритмами, которые необходимы ему для работы. В следующем разделе более подробно изучается одна из подсистем космического аппарата – система ориентации и стабилизации.

P. S.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru

ЧАСТЬ IV

ИСПОЛНИТЕЛЬНЫЕ УСТРОЙСТВА

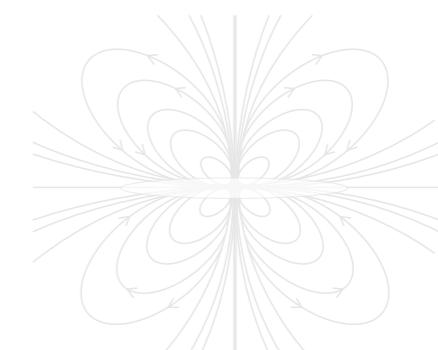
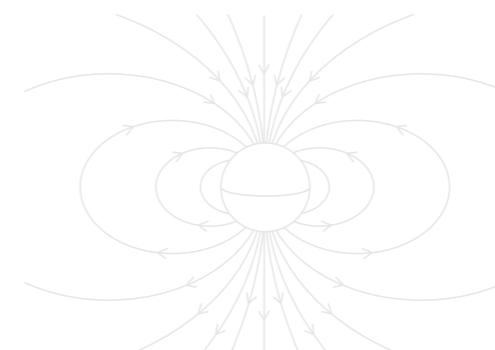
МОДУЛЬ ПРЕДНАЗНАЧЕН ДЛЯ ПРЕДОСТАВЛЕНИЯ УЧЕНИКАМ ЗНАНИЙ О ПРИНЦИПАХ СТАБИЛИЗАЦИИ И ОРИЕНТАЦИИ КОСМИЧЕСКИХ АППАРАТОВ. ЗАДАЧЕЙ МОДУЛЯ ТАКЖЕ ЯВЛЯЕТСЯ РАССМОТРЕНИЕ ИСПОЛНИТЕЛЬНЫХ УСТРОЙСТВ.

ТЕОРЕТИЧЕСКИЕ ЗНАНИЯ ПОМОГАЮТ ОЗНАКОМИТЬСЯ С ИСПОЛНИТЕЛЬНЫМИ УСТРОЙСТВАМИ ОРИЕНТАЦИИ КОСМИЧЕСКОГО АППАРАТА. РЕЧЬ ИДЕТ О ПРИБОРАХ, АГРЕГАТАХ, КОНСТРУКЦИОННЫХ ОСОБЕННОСТЯХ КОСМИЧЕСКИХ АППАРАТОВ, ИСПОЛЬЗУЕМЫХ ДЛЯ ОБЕСПЕЧЕНИЯ ПРАВИЛЬНОГО ПОЛОЖЕНИЯ В ПРОСТРАНСТВЕ.

НА ОРБИТЕ НЕВОЗМОЖНО ИСПОЛЬЗОВАТЬ ПРИВЫЧНЫЕ ДЛЯ НАЗЕМНЫХ УСТРОЙСТВ МЕТОДЫ И СРЕДСТВА ПЕРЕДВИЖЕНИЯ: НЕ ОТ ЧЕГО ОТТОЛКНУТЬСЯ. ОДНАКО ДАЖЕ В ТАКИХ УСЛОВИЯХ МОЖНО ОРИЕНТИРОВАТЬСЯ, МЕНЯТЬ СВОЕ ПОЛОЖЕНИЕ В ПРОСТРАНСТВЕ ЗА СЧЕТ ТРАНСФОРМИРОВАНИЯ ВНУТРЕННЕЙ ЭНЕРГИИ В ДВИЖЕНИЕ ИЛИ ВЗАИМОДЕЙСТВИЯ С МАГНИТНЫМ И ГРАВИТАЦИОННЫМ ПОЛЯМИ ЗЕМЛИ.

ТЕОРЕТИЧЕСКИЕ ЗНАНИЯ НАГЛЯДНО ДЕМОНСТРИРУЮТ ПРИМЕНЕНИЕ ИЗУЧАЕМЫХ НА УРОКАХ ФИЗИКИ ТЕОРИИ МАГНИТНОГО ПОЛЯ И ЗАКОНА СОХРАНЕНИЯ КИНЕТИЧЕСКОГО МОМЕНТА.

ПРАКТИЧЕСКАЯ ЧАСТЬ НАГЛЯДНО ДЕМОНСТРИРУЕТ СПОСОБЫ КОНТРОЛЯ ОРИЕНТАЦИИ РЕАЛЬНОГО КОСМИЧЕСКОГО АППАРАТА.



Теоретические основы

ГРУППОВАЯ
РАБОТА

2 ЧАСА

Способы ориентации космических аппаратов

РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ:

Рекомендуется задействовать фантазию ученика и совместно с ним определить, на что же может «опереться» космический аппарат, находящийся на орбите Земли. Это может быть атмосфера, магнитное поле планеты, кинетические эффекты. Когда ученики предложат один или несколько факторов, предлагается подробнее рассмотреть один из них, позволив ученику самостоятельно придумать, как ориентировать аппарат с использованием разбираемого фактора: например, как используется разреженная атмосфера при орбитальном полете. При таком подходе лучше использовать диалоговый режим теоретической лекции, поддерживающий активность ученика.

ЦЕЛЬ ЗАНЯТИЯ

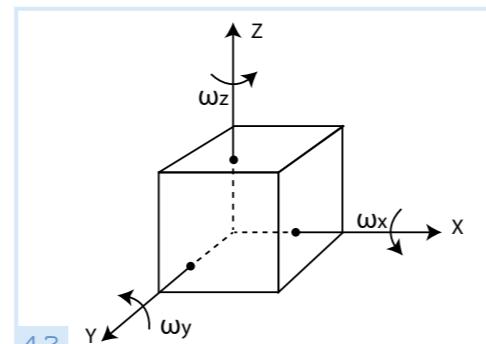
Теоретическое занятие позволит создать необходимую базу знаний, опираясь на которую, ученик получит представление об условиях, в которых эксплуатируется космический аппарат, и которые обеспечивают ему требуемую ориентацию и стабилизацию. На этом занятии будут подробно рассмотрены различные методы ориентации космических аппаратов. Знания применяются в практических блоках данного модуля.

ВВЕДЕНИЕ И ТЕОРИЯ

Наверное, все представляют себе первый спутник, выведенный на орбиту 4 октября 1957 года (рис. 4.1). Но не часто мы задумываемся, как он был ориентирован в процессе полета. Неверно полагать, что он двигался как воланчик, антенны назад.



4.1



4.2

ЧТО ЗНАЧИТ «ОРИЕНТИРОВАТЬ КОСМИЧЕСКИЙ АППАРАТ»?

Изменение ориентации космического аппарата без изменения параметров орбиты может происходить только за счет модификации угловых скоростей вращения вокруг связанных осей X, Y, Z (рисунок 4.2).

4.1 ПЕРВЫЙ ИСКУССТВЕННЫЙ СПУТНИК ЗЕМЛИ ПС-1 (ПРОСТЕЙШИЙ СПУТНИК-1)

4.2 ОСИ СВЯЗАННОЙ СИСТЕМЫ КООРДИНАТ

«ОТТАЛКИВАНИЕ» СПУТНИКА

От чего может «оттолкнуться» космический аппарат, чтобы приобрести нужные угловые скорости? На орбите космический аппарат может:

- использовать накопленную внутри себя энергию,
- взаимодействовать с атмосферой Земли,
- взаимодействовать с солнечным излучением,
- взаимодействовать с полями, в которых он находится: гравитационное и магнитное поля Земли.

Выделяют активные и пассивные методы ориентации космических аппаратов, о которых пойдет речь ниже.

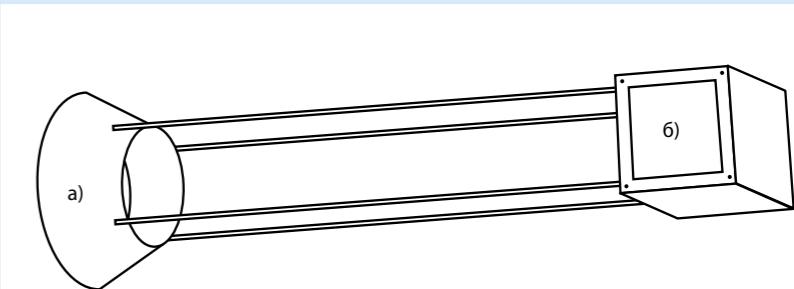
ПАССИВНЫЕ МЕТОДЫ ОРИЕНТАЦИИ КОСМИЧЕСКОГО АППАРАТА

ВЗАИМОДЕЙСТВИЕ С АТМОСФЕРОЙ ЗЕМЛИ

Казалось бы, спутники летают в безвоздушном пространстве, однако, всем известно, что орбиту Международной космической станции (МКС) периодически поднимают. Опускается она как раз из-за влияния атмосферы, трения о разреженный воздух, который присутствует на высоте 400 км.

Для объекта, движущегося в потоке среды, важны два параметра: плотность среды и скорость передвижения. Например, бежать по беговой дорожке со скоростью 10 км/ч гораздо легче, чем с такой же скоростью плыть в воде; самолет имеет некоторую минимальную скорость, при которой подъемная сила крыла создает достаточную для горизонтального полета подъемную силу. Иными словами, можно «зависнуть» на высоте 400 км и не ощущать на себе влияние атмосферы, но можно лететь там со скоростью 7,6 км/с (линейная скорость МКС) и ощущать легкое сопротивление атмосферы. Итак, где же заканчивается атмосфера Земли? Правильно ответить: там где ее влияние пренебрежимо мало, поэтому нужно рассматривать этот вопрос отдельно для каждого случая.

Если влияние атмосферы на космический аппарат пренебрежимо мало, то форма аппарата не имеет значения для его орбитального движения. Но если атмосфера оказывает достаточное влияние на космический аппарат, то, создавая определенную форму аппарата (например, наподобие воланчика) можно ориентировать его вдоль направления воздушного потока. Пример космического аппарата с аэродинамической составляющей



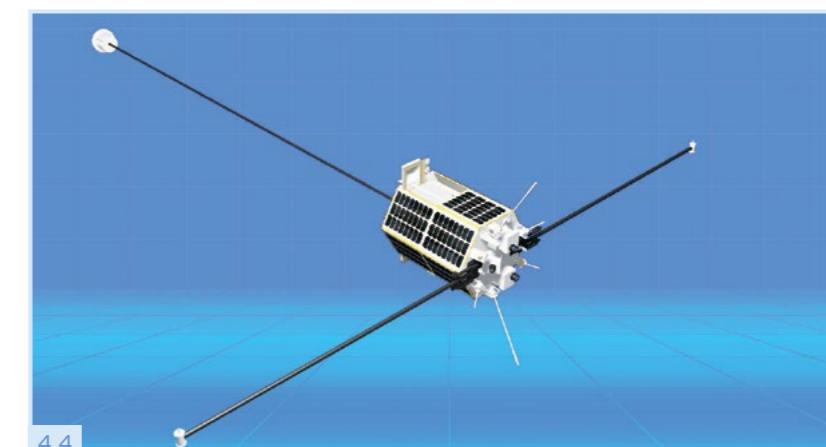
4.3

щай представлен на рисунке 4.3.

ВЗАИМОДЕЙСТВИЕ С ГРАВИТАЦИОННЫМ ПОЛЕМ ЗЕМЛИ

Любое тело, имеющее массу, обладает гравитационным полем. Причем сила, с которой одно тело притягивает второе, прямо пропорциональна произведению масс тел и обратно пропорциональна квадрату расстояния между телами. Все мы ощущаем силу притяжения Земли и знаем, что она равна произведению массы нашего тела на ускорение свободного падения. Запишем второй закон Ньютона для тела, находящегося на орбите высотой $h=400$ км и имеющего массу $m=420$ кг:

$$\Sigma F = ma \Leftrightarrow mg = ma_{\text{ц}} \Leftrightarrow mg = mV^2 \Leftrightarrow V = \sqrt{gR}$$



4.4

Полученная формула скорости есть первая космическая скорость тела на орбите радиусом:

$$R = R_{\text{земли}} + h, \text{ км, где: } R_{\text{земли}} = 6371 \text{ км.}$$

Обратите внимание: в формуле отсутствует масса аппарата. Два тела разной массы, находящиеся на одной орбите, будут обладать разным количеством движения (импульсом). Заметим, что ускорение свободного падения незначительно меньше для такой высоты по сравнению с земной.*

4.3 ПРИНЦИПИАЛЬНАЯ СХЕМА СПУТНИКА С КОНЫСНЫМ АЭРОДИНАМИЧЕСКИМ СТАБИЛИЗАТОРОМ
а) стабилизатор;
б) спутник.

4.4 КОСМИЧЕСКИЙ АППАРАТ «ЮБИЛЕЙНЫЙ» С ГРАВИТАЦИОННОЙ ШТАНГОЙ

* Предлагается показать самостоятельно.

Теперь проще понять, как работает гравитационная ориентация. Спутники с такой системой имеют длинную штангу с грузом на конце. Можно рассматривать такой космический аппарат как «гантель», на противоположные концы которой будут действовать разные по модулю силы притяжения.

На рисунке 4.4 представлен космический аппарат «Юбилейный», созданный ОАО «ИСС» им. академика М. Ф. Решетнёва совместно с группой российских космических предприятий и высших учебных заведений. Особенности такого типа ориентации:

- низкая точность ориентации;
- продолжительное время перехода в стабилизированный режим;
- ориентация возможна только вокруг одной оси, проходящей через центр Земли;
- для низких орбит могут быть отклонения, связанные с взаимодействием с верхними слоями атмосферы Земли.

Наиболее интересные космические аппараты, имеющие такую систему ориентации: «Юбилейный», «Рубин – 4», «Гонец-М», «Триад», «Траак», «GEOS-1, -2», «Эол», спутники серии ATS, «Эксплорер-38» и другие.*



4.5 АКТИВНЫЕ МЕТОДЫ ОРИЕНТАЦИИ КОСМИЧЕСКОГО АППАРАТА

ОРИЕНТАЦИЯ КОСМИЧЕСКОГО АППАРАТА С ПОМОЩЬЮ РЕАКТИВНЫХ ДВИГАТЕЛЕЙ

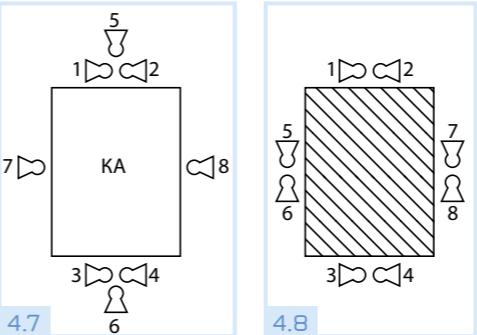
Самый очевидный способ ориентировать космический аппарат на орбите – установить на него реактивные двигатели. Причем применяются как газореактивные двигатели, использующие баллоны высокого давления (например, сжатый азот), так и химические ракетные двигатели, чаще всего однокомпонентные (например,

использующие пероксид водорода или производные гидразина). Этот метод широко применяется в пилотируемой космонавтике. На всех советских космических кораблях: «Восток», «Восход», «Союз» (рисунок 4.5) для возвращения на Землю необходим тормозной импульс. Для построения правильной ориентации перед выдачей импульса используются двигатели ориентации, которые разворачивают космический аппарат так, чтобы сопло тормозного двигателя было направлено против движения космического корабля. Ракетные двигатели ориентации установлены на пилотируемых лунных космических кораблях, а также на крылатых аппаратах: «Space Shuttle» и «Буран» (рисунок 4.6).

По сути, химические ракетные двигатели ориентации являются уменьшенным вариантом двигателей, применяемых в ракетах-носителях. Химические двигатели используются и для ориентации среднеразмерных спутников, например «Deep Space 1». В том числе, химические ракетные двигатели применяют для разгрузки насыщенных двигателей-маховиков, речь о которых пойдет в следующем разделе.

Помимо химических, в настоящее время существует достаточно много разновидностей двигателей ориентации: электрические,

* Предлагается самостоятельно ознакомиться с конструкцией данных аппаратов.



двигатели используются в качестве маршевых, т. е. увеличивающих линейную скорость аппарата. Но иногда они применяются и для ориентации космических аппаратов. При этом двигатели могут работать по схеме на рисунке 4.7: крен обеспечивает двигатели 1+4 и 2+3, поступательное движение – 5+6 и 7+8. На рисунке 4.8: управление по крену производится с помощью попарно работающих двигателей 1+4 и 2+3, или 5+8, 6+7, поступательное движение – с помощью 1+3 и 2+4 или 5+7 или 6+8.

ВЗАИМОДЕЙСТВИЕ С МАГНИТНЫМ ПОЛЕМ ЗЕМЛИ

Все видели компас: стрелка стремится повернуться в направлении Северного полюса, точнее, северного магнитного полюса, который на самом деле не совпадает с северным географическим полюсом. Вдобавок, магнитные полюса находятся в постоянном движении, смещаясь со скоростью примерно 10 км в год. По одной из теорий, существование магнитного поля объясняется процессами, происходящими в недрах планеты: раскаленные металлы движутся, перемешиваются, а как известно, поток движущихся заряженных частиц порождает магнитное поле. Этим и объясняется неоднородность поля и дрейф полюсов.

Спутник может вести себя аналогично стрелке компаса, только вместо постоянного магнита используют катушки индуктивности (рис. 4.9) или их частный случай – соленоиды (рис. 4.10). Соленоид – обязательно цилиндрическая катушка из проводника,

причем длина такой катушки многократно превышает ее диаметр. Если требуется обеспечить спутнику определенные угловые скорости, ставят три соленоида вдоль трех осей связанной со спутником декартовой системы координат. Тогда, пропуская ток определенной величины и направления через витки катушек, можно добиться того, что собственный вектор магнитной индукции спутника определенным образом позиционируется относительно местного магнитного поля Земли.

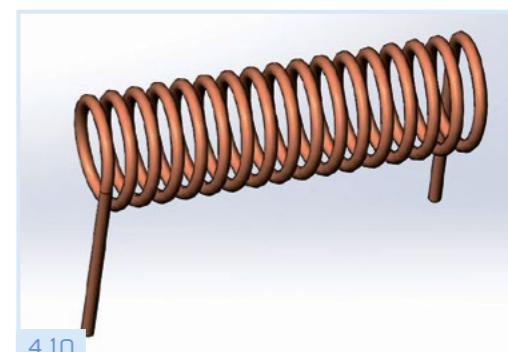
Преимущества представленного метода ориентации: отсутствие движущихся механизмов, простота устройства, небольшая масса и возможность использовать энергию солнечных батарей. Из минусов отметим низкую точность и небольшие величины воздействующих моментов, которые не позволяют применять данный метод для низкоорбитальных космических аппаратов. Из-за низкой точности электромагнитные системы используют для разгрузки двигателей-маховиков при их насыщении. Еще одним ограничением является наличие магнитного поля. Действие магнитного поля Земли распространяется примерно на 60 тыс. км, а дальше – уже межпланетное пространство. В Солнечной системе магнитным полем обладают только Земля и планеты-гиганты: Юпитер, Сатурн, Уран, Нептун. Причем магнитные поля Урана и Нептуна являются не дипольными, а квадрупольными, т.е. имеют по 2 северных и 2 южных полюса (см. рисунок 4.11)**.

4.7 4.8 СХЕМЫ РАСПОЛОЖЕНИЯ РЕАКТИВНЫХ ДВИГАТЕЛЕЙ ОРИЕНТАЦИИ КОСМИЧЕСКОГО АППАРАТА

** В данном модуле ученику предстоит самостоятельно наблюдать за процессом переориентации магнитной катушки, помещенной в магнитное поле.



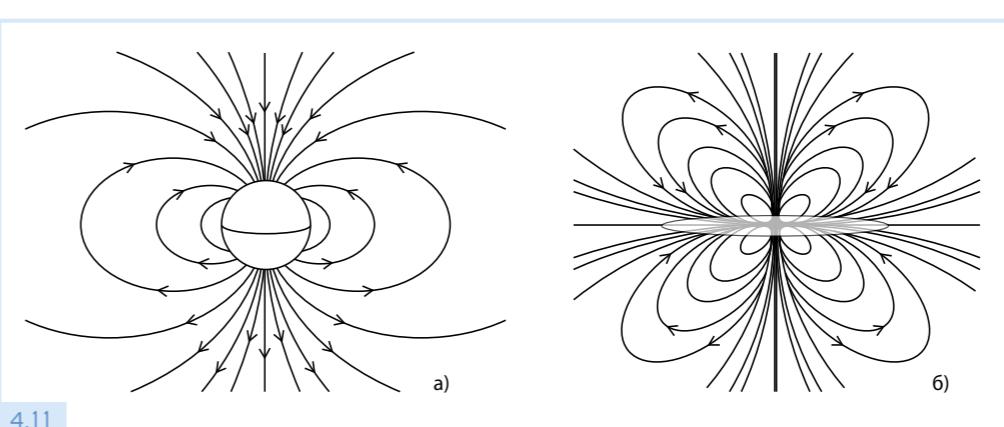
4.9



4.10

ЭЛЕМЕНТЫ ЭЛЕКТРОМАГНИТНОЙ СИСТЕМЫ СТАБИЛИЗАЦИИ:

4.9 Катушка индуктивности;
4.10 Соленоид



ЭЛЕМЕНТЫ ЭЛЕКТРОМАГНИТНОЙ СИСТЕМЫ СТАБИЛИЗАЦИИ:

4.11 СХЕМАТИЧЕСКОЕ ИЗОБРАЖЕНИЕ МАГНИТНЫХ ПОЛЕЙ:
а) Дипольное;
б) Квадрупольное

* В данном модуле этот метод ориентации представлен практическими занятиями.

ОРИЕНТАЦИЯ ЗА СЧЕТ ЭФФЕКТА СОХРАНЕНИЯ КИНЕТИЧЕСКОГО МОМЕНТА

В данном разделе можно выделить три типа: стабилизация вращением, двигатели-маховики и гиродины. Для того, чтобы связать практические представления об этом типе стабилизации с теорией, представим себе юлу или волчок. В земных условиях они рано или поздно остановятся. Это связано с взаимодействием тела с тормозящей процесс вращения силой трения о стол, на котором происходит вращение, а если же он происходит на магнитном подвесе – то об атмосфере. Если исключить влияние различных опор и молекул воздуха, тело будет вращаться бесконечно долго. Этот процесс описывается **законом сохранения кинетического момента**.

Для его понимания введем несколько терминов. Главный вектор сил, который будет являться векторной суммой всех сил системы. Главный момент количества движения (импульсов) – векторная сумма моментов всех импульсов системы относительно некоего центра или в нашем частном случае, относительно оси. Тогда закон вращения твердого тела вокруг оси будет звучать так: «Если главный момент внешних сил системы относительно выбранной оси равен нулю, то главный момент количества движения относительно этой оси будет постоянным». Или так: «Изолированное от внешних моментов

абсолютно твердое тело будет сохранять состояние покоя или равномерного вращения вокруг неподвижной оси до тех пор, пока приложенные к этому телу внешние силы и моменты не заставят его изменить это состояние».

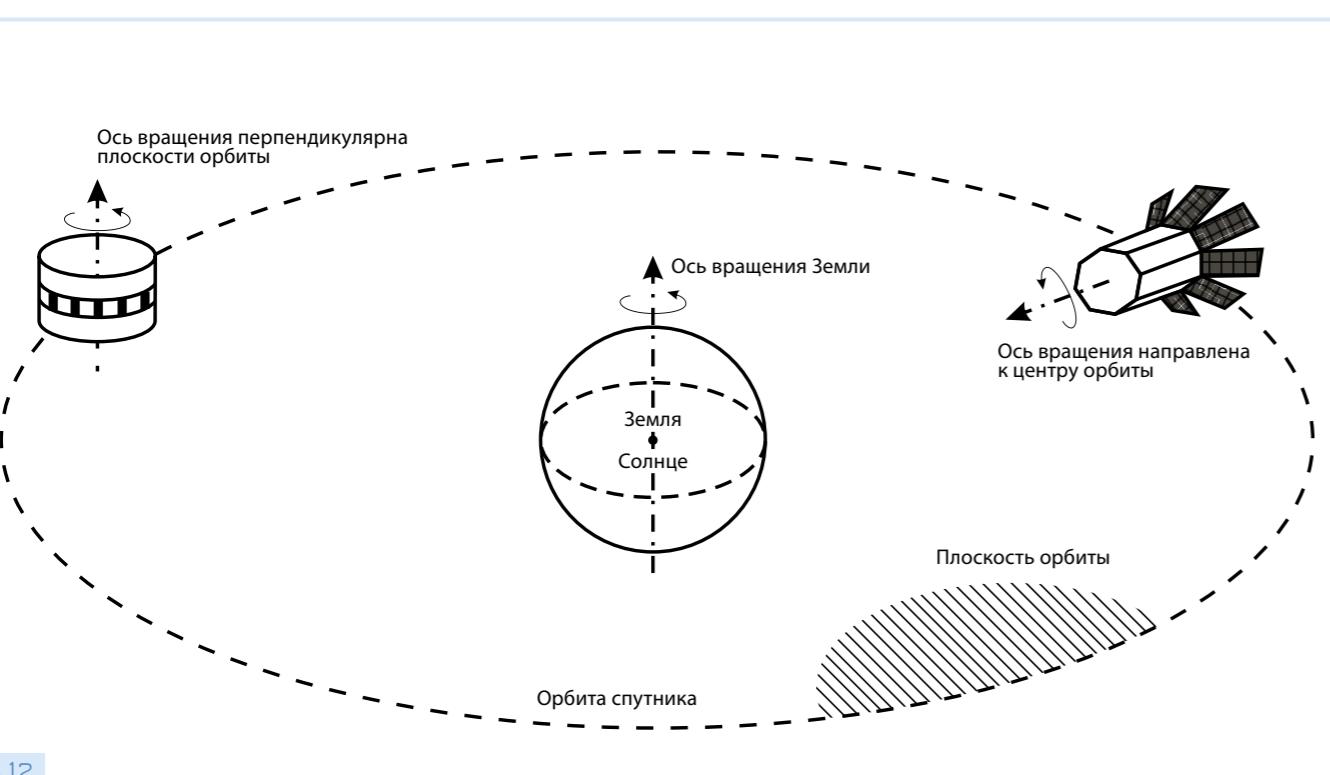
ОРИЕНТАЦИЯ ЗА СЧЕТ СТАБИЛИЗАЦИИ ВРАЩЕНИЕМ

Этот тип ориентации можно назвать самым первым: в начале космической эры аппараты раскручивали вдоль продольной оси, как показано на рисунке 4.12. Тогда ориентация в пространстве будет сохраняться на орбите сколь угодно долго. Так все первые космические аппараты США были стабилизированы вращением.

Важно обеспечить вращение относительно продольной оси аппарата, проходящей через его центр масс, иначе ось вращения будет двигаться: волчок ведет себя аналогично на самых первых секундах вращения и на самых последних. Только волчок, в отличие от космического аппарата, имеет точку опоры – точку касания ножки стола, без которой ось вращения не принимала бы каждый раз строго вертикальное положение.

Спутник, которому сообщили достаточно большую скорость, несложно затормозить. Достаточно выдвинуть штанги симметрично, в направлении от оси вращения, тем самым увеличив момент инерции вращающегося тела. Напомним,

4.12 ОРИЕНТАЦИЯ СПУТНИКОВ, СТАБИЛИЗИРОВАННЫХ ВРАЩЕНИЕМ



4.12

что моментом инерции механической системы относительно неподвижной оси называется величина J , равная сумме произведений масс всех n материальных точек системы на квадраты их расстояний до оси:

$$J = \sum_{i=1}^n m_i r_i^2$$

m_i – масса i -й точки;

r_i – расстояние от i -й точки до оси.

§ Предлагается самостоятельно показать, почему при увеличении момента инерции упадет угловая скорость вращения.

МАХОВИКИ И ГИРОДИНЫ

С самого начала отметим, что гироскоп – прибор, показывающий угловую скорость вращения тела, а маховик и гиродины очень на него похожи внешне, но по сути являются другими приборами. Итак, действительно, пусть в нашем космическом аппарате на орбите установлен вращающийся диск. Ось вращения диска проходит через центр масс спутника. Пусть в самом начале спутник и диск неподвижны. Если мы начнем раскручивать электромотором диск внутри спутника по часовой стрелке, то спутник начнет вращаться в противоположную сторону. Когда мы остановим вращение диска относительно космического аппарата, его вращение также прекратится. Почему так происходит?

Причина – законы сохранения. В данном случае работает закон сохранения момента количества движения (импульса). Ось маховика жестко закреплена. Ориентация космического аппарата обеспечивается регулированием скорости вращения маховика. Для создания трехосевой ориентации необходимы три маховика. Механические различия между двигателем-маховиком и гиродином заключаются в том, что ось гиродина не закреплена, а находится в рамке (рисунок 4.13) анало-

гично закреплению колеса велосипеда: свойство маховика удерживать положение вращающейся оси не позволяет нам падать.*

Преимуществом маховиков и гиродинов является их потребность в электрической энергии, получаемой с солнечных батарей; топливо не расходуется. Однако минусом является сложность конструкции, наличие вращающихся элементов и ограничение скоростей вращения. У каждого двигателя-маховика есть предел угловой скорости, которую он может развить. Если на космический аппарат периодически действует сила, закручивающая его в одну и ту же сторону, такой предел рано или поздно наступит. Когда маховик достигает предельных скоростей вращения, наступает «эффект насыщения», который заключается в том, что, при достижении определенного значения угловой скорости из-за электрических эффектов, ток будет стремиться к нулю. В таких случаях маховик старается «разгрузить». Разгрузжение происходит за счет других двигателей ориентации: электромагнитных систем, реактивных двигателей.

Гиродины используют и на Международной космической станции – четыре блока с размерами более метра по осям измерений и массой около 300 кг. В данном модуле этот метод ориентации представлен практическими занятиями.

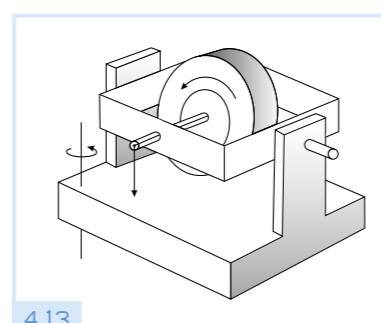
ЧТО ДАЛЬШЕ?

Рекомендуется выполнить задания для самостоятельной работы, указанные в описании занятия. Подробнее изучить космические аппараты, использующие интересные ученику способы ориентации, для чего можно сделать макет своими руками (классическое моделирование, 3D-печать).

P. S.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru

4.13 УСТРОЙСТВО ГИРОДИНА



4.13

4.14 ОБЩИЙ ВИД ДВУХСТЕПЕННОГО ГИРОСКОПА



4.14

Практика I



РАБОТА В МАЛЫХ ГРУППАХ



2 ЧАСА

Исследование стабилизации с использованием двигателя-маховика

С набором, как и в предыдущих занятиях, может работать от двух до четырех человек. Постарайтесь сделать так, чтобы учащиеся распределили обязанности как можно более равномерно. Например, пока кто-то занимается пайкой, кто-то может собирать основную часть корпуса, а кто-то разбираться с работой датчиков.

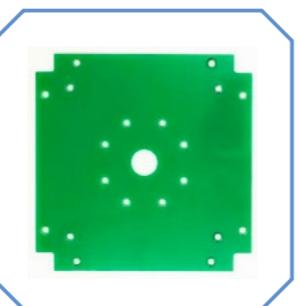
ЦЕЛЬ ЗАНЯТИЯ

Сборка макета наноспутника, близкого к форм-фактору «Кубсат 1U». В макет входит подсистема стабилизации, состоящая из двигателя, маховика и датчика угловой скорости.

ВВЕДЕНИЕ

Магнитные катушки применяются на большом количестве аппаратов. Как упоминалось ранее, они могут быть использованы и как самостоятельная система, и в роли вспомогательной. Двигатели-маховики необходимо разгружать, на случай если накопленный ими кинетический момент достигнет предельного значения. К тому же, иметь на борту космического аппарата прибор, в котором находится быстро вращающийся механический элемент, ненадежно. Однако к настоящему моменту использование дви-

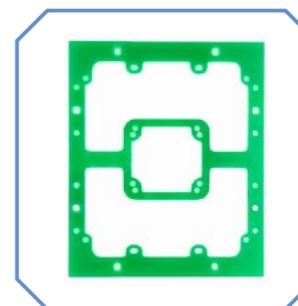
МАТЕРИАЛЫ И КОМПОНЕНТЫ



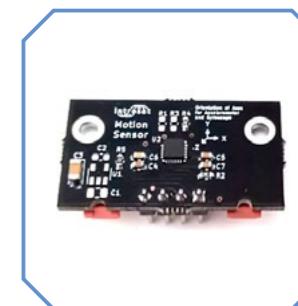
НИЖНЯЯ ПАНЕЛЬ, 1 ШТ.



ВЕРХНЯЯ ПАНЕЛЬ, 1 ШТ.



БОКОВАЯ ПАНЕЛЬ, 4 ШТ.



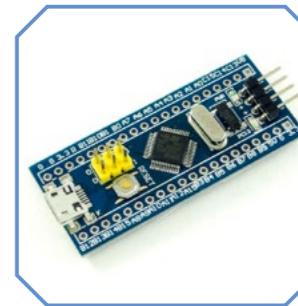
ДАТЧИК ПОЗИЦИОНИРОВАНИЯ, 1 ШТ.



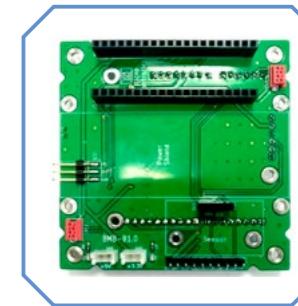
ПЛАТА УПРАВЛЕНИЯ МАХОВИКОМ, 1 ШТ.



МАХОВИК, 1 ШТ.



ПЛАТА УПРАВЛЕНИЯ BLUE PILL, 1 ШТ.



МАТЕРИНСКАЯ ПЛАТА, 1 ШТ.



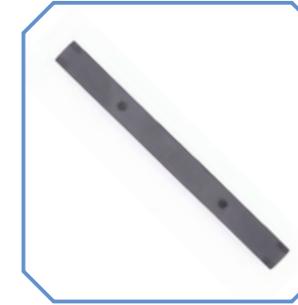
РАЗЪЕМЫ MICROMATCH НА ШЛЕЙФ, 2 ШТ.



РЕЛЬС, 4 ШТ.



ПЕРЕКЛАДИНА С ПАЗОМ, 4 ШТ.



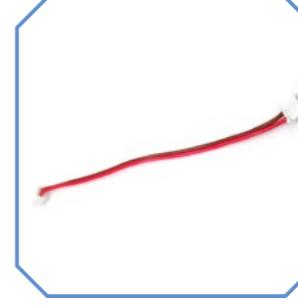
ПЕРЕКЛАДИНА С ТОРЦЕВЫМ ОТВЕРСТИЕМ, 4 ШТ.



МОДУЛЬ ПИТАНИЯ С АККУМУЛЯТОРОМ, 1 ШТ.



НАБОР МОНТАЖНЫХ ПРОВОДОВ



ПРОВОД ПИТАНИЯ, 1 ШТ.



ВЕРТЛЮГ С ПОДВЕСНЫМ ЭЛЕМЕНТОМ, 1 ШТ.



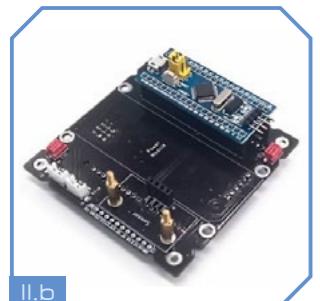
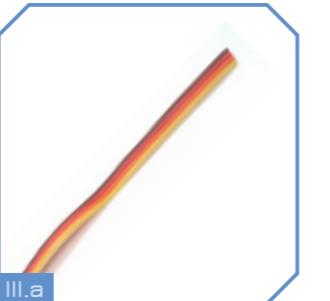
4.15 ВИД УСТАНОВКИ



НАБОР СТОЕК ДЛЯ ПЕЧАТНЫХ ПЛАТ M2.5



НАБОР КРЕПЕЖА M2.5



СБОРКА МИКРОСПУТНИКА

Сборка корпуса приводилась в занятиях предыдущего раздела и в данном случае ничем не отличается.

I. Крепление двигателя-маховика

I.a Установите маховик на адаптер на оси двигателя с помощью пары винтов M2.5x6 с выпуклой головкой.

I.b Закрепите плату на нижней панели с помощью винтов M2.5x12 с потайной головкой, используя дополнительные гайки в качестве прокладок между панелью и платой.

I.c Прикрепите нижнюю панель к раме с помощью винтов M2.5x6 с потайной головкой.

II. Сборка системы управления

II.a Установите плату BluePill на материнскую плату.

II.b-c Установите датчик позиционирования на материнскую плату с помощью пары стоек 11мм, винтов M2.5x6 с полукруглой головкой и гаек. Перед установкой обязательно проверьте, чтобы переключатель на обратной стороне датчика был в положении «ON».

II.d Закрепите 4 стойки PCHSN5 (длиной 5 мм) на материнской плате при помощи винтов M2.5x4 с полукруглой головкой.

II.e Установите аккумулятор в модуль питания, при необходимости закрепите его

пластиковой стяжкой через ушки в плате. Модуль питания установите на материнскую плату так, чтобы шестиконтактный разъемы сопрягались друг с другом. Закрепите гайками модуль питания.

II.f Установите модуль Bluetooth в соответствующий разъем (если планируете использовать беспроводное подключение).

III. Обжим шлейфа для I2C

III.a Разделите многопроводной шлейф на шлейфы по 4 провода.

III.b Вставьте шлейф в разъем. Обратите внимание, что разъемы MicroMatch ориентированные, то есть с одной стороны снабжены выступом, для которого на плате есть ответное отверстие, таким образом собранный шлейф можно подключить только одним способом.

III.c Расположите разъем со шлейфом на кримпере, так чтобы разъем находился посередине рабочей части кримпера, а выступающая часть разъема расположилась в одной из прорезей вспомогательной детали.

III.d Сожмите ручки кримпера. При этом действие должно послышаться тихий щелчок, означающий, что разъем обжался. Аналогичным образом обожмите разъем на втором конце шлейфа. При обжиме второго разъема следите за тем, чтобы разъемы были расположены одинаково: выступающие части разъемов должны находиться по одну сторону от шлейфа.

IV. Финальная сборка

IV.a Подключите модуль маховика к материнской плате с помощью шлейфа I2C и провода питания. Для этого можно использовать любые из доступных разъемов MicroMatch на материнской плате и модуле маховика. Питание на двигатель маховика подается от 5-вольтового вывода материнской платы. Установка боковых панелей в этой задаче не требуется.

ПРОВЕРКА РАБОТОСПОСОБНОСТИ

После сборки устройства необходимо проверить его работоспособность. По сути, во всей собранной системе два управляемых устройства: двигатель и датчик позиционирования.

И датчик позиционирования, и плата управления двигателем работают по шине I2C. Шина I2C позволяет подключить одновременно до 127 устройств, которые смогут управляться от одного контроллера. Управляющее устройство шины также принято называть master, а зависимые – slave. В нашем случае плата Blue Pill – master, а датчик позиционирования – slave.

Шина I2C состоит из двух линий: SCL и SDA. Шина SCL является линией тактирования и отвечает за синхронизацию устройств между собой по частоте работы. Шина SDA является шиной данных и отвечает за их прием и передачу.

Для работы с шиной I2C в Arduino IDE используется библиотека Wire*, которая позволяет микроконтроллеру обращаться

к зависимым устройствам и осуществлять прием и передачу данных.

Каждое устройство, подключенное к шине I2C имеет свой адрес. Обычно данный адрес задается производителем устройства и указывается в datasheet устройства. В нашем случае, адрес датчика – 0x68, а адрес двигателя – 0x33.

Чтобы прослушать шину и узнать, по каким адресам подключены устройства, можно использовать код, приведенный далее.

Код для поиска устройств, подключенных к шине I2C:

```
#include <Wire.h>
#include <IS_Bluetooth.h>

void setup(){
    Wire.begin();
    Serial.begin(115200, SERIAL_8E1);
    Serial.println("I2C Scanner");
}

void loop(){
    if(Serial.available()){
        char command = Serial.read();
        if(command == 'b'){
            enter_bootloader();
        }
    }
}
```

* Более подробно о библиотеке Wire можно прочитать здесь: <https://all-arduino.ru/biblioteki-arduino/arduino-biblioteka-wire/>.

** Библиотеку «IS_Bluetooth» и другие библиотеки можно скачать по ссылке: introsat.ru/download

```

byte error, address;
int nDevices;

Serial.println("Scanning...");

nDevices = 0;
for (address = 8; address < 127; address++) {
  Wire.beginTransmission(address);
  delay(100);
  error = Wire.endTransmission();
}

if(error == 0) {
  Serial.print("I2C device found at address 0x");
  if(address < 16)
    Serial.print("0");
  Serial.print(address, HEX);
  Serial.println("!");
}

nDevices++;
}

else if(error == 4) {
  Serial.print("Unknown error at address 0x");
  if(address < 16)
    Serial.print("0");
  Serial.println(address, HEX);
}

if(nDevices == 0)
  Serial.println("No I2C devices found\n");
else
  Serial.println("done\n");

delay(1000);
}

```

Если все правильно, то в выводе монитора порта появится два адреса: 0x33 и 0x68.

```

Scanning...
I2C device found at address 0x33 !
I2C device found at address 0x68 !
done

```

4.16

ПРИМЕЧАНИЕ. Адрес двигателя может не определяться, если к нему не подключен кабель питания, а плата питания находится в выключенном состоянии.

Управление двигателем реализовано пошине I2C. Однако в зависимости от версии платы управления маховиком, управление им может отличаться. Версию платы можно найти на самой плате.

Начиная с версии BFM1.75, используется расширенная версия управления, включающая в себя помимо канала управ-



	Версии до BFM1.75	Версии с BFM1.75
Канал управления катушками	Нет	Да
Переключение I2C-адресов	Нет	Да
Возможность отключения/подключения световой индикации	Нет	Да
Подключаемая библиотека	IS_Drive_1	IS_Drive_2
Функция управления	<p>speed_control(uint8_t addr, uint8_t dir, uint16_t mot_speed)</p> <p>Функция в качестве входных данных использует следующие данные:</p> <ul style="list-style-type: none"> addr – адрес устройства, по умолчанию 0x33; dir – направление вращения маховика, 0 задает вращение по часовой стрелке, 1 – против часовой; mot_speed – скорость вращения двигателя (принимает значения от 0 до 6000). 	<p>speed_control(uint8_t addr, uint8_t dir, uint16_t mot_speed, uint8_t channel)</p> <p>Функция в качестве входных данных использует следующие данные:</p> <ul style="list-style-type: none"> addr – адрес устройства, по умолчанию 0x33; dir: <ul style="list-style-type: none"> для маховика: направление вращения, 0 задает вращение по часовой стрелке, 1 – против часовой; для катушек: полярность выхода, 0 задает одну полярность, 1 – другую; mot_speed: <ul style="list-style-type: none"> для маховика: скорость вращения двигателя (принимает значения от 0 до 6000); для катушек: регулирует выходное значение ШИМ (принимает значения от 0 до 65535) channel – задает канал управления, 0 – управление маховиком, 1 – управление катушками.

ления маховиком также канал управления катушками.

Чтобы проверить работоспособность двигателя, необходимо задать ему управление. Пример кода для скетча, задающего вращение двигателя, приведен далее. Если

4.16 ОПРЕДЕЛЕНИЕ I2C-АДРЕСОВ ДВИГАТЕЛЯ И ДАТЧИКА ПОЗИЦИОНИРОВАНИЯ

при загрузке скетча двигатель постепенно разгоняется, а затем останавливается и меняет направление вращения, то устройство работает корректно. В противном случае необходимо проверить возможные ошибки в соединении шлейфов и в обжиме самих шлейфов, а также убедиться, что к плате маховика подключен шлейф питания, идущий от материнской платы.

Код для двигателя версии до BFM1.75:

```

//Подключаем библиотеку для работы с маховиком
#include <IS_Drive.h>
#include <IS_Bluetooth.h>

//Задаем I2C-адрес двигателя
#define IS_Drive_1 0x33

//Задаем объект класса Drive
Drive drive;

//Задаем переменную угловой скорости
uint16_t MotSpeed = 0;

void setup() {
  //Инициализируем работу I2C
  Wire.begin();
  //Инициализируем работу UART
  Serial.begin(115200, SERIAL_8E1);
}

void loop() {
  char msg = Serial.read();
  if(msg == 'b'){
    enter_bootloader();
  }
  //Постепенно наращиваем скорость вращения против часовой стрелки
  while(MotSpeed < 3000){
    //Увеличиваем скорость на 50 об/мин на каждой итерации цикла while
    MotSpeed += 50;
  }
}

```

Код для двигателя версии с BFM1.75:

```

//Подключаем библиотеку для работы с маховиком
#include <IS_Drive_Two_Ch.h>
#include <IS_Bluetooth.h>

//Задаем I2C-адрес двигателя
#define IS_Drive_1 0x33

//Задаем объект класса Drive
Drive drive;

//Задаем переменную угловой скорости
uint16_t MotSpeed = 0;

void setup() {
  //Init();
  //Инициализируем работу I2C
  Wire.begin();

  //Инициализируем работу UART
  Serial.begin(115200, SERIAL_8E1);
}

void loop() {

```

```

char msg = Serial.read();
if (msg == 'b') {
    enter_bootloader();
}

//Постепенно наращиваем скорость
//вращения против часовой стрелки
while (MotSpeed < 3000)
{
    //Увеличиваем скорость на 50 об/мин на
    //каждой итерации цикла while
    MotSpeed += 50;
    //Функция, задающая направление и
    //скорость вращения
    drive.speed_control(lS_Drive_1, 1, MotSpeed,
0);
    //Выводим текущую скорость в UART
    Serial.println(MotSpeed);
    //Задержка между итерациями для
    //плавного уменьшения угловой скорости
    delay(200);
}

//Функция, задающая направление и
//скорость вращения
drive.speed_control(lS_Drive_1, 0, MotSpeed,
0);
//Выводим текущую скорость в UART
Serial.println(MotSpeed);
//Задержка между итерациями для
//плавного увеличения угловой скорости
delay(200);
}

//Обнуляем переменную MotSpeed
MotSpeed = 0;
//Останавливаем маховик
drive.speed_control(lS_Drive_1, 0, MotSpeed,
0);
//Даем маховику немного времени на
//остановку
delay(4000);

//Постепенно наращиваем скорость
//вращения по часовой стрелке

```

ЧТО ДАЛЬШЕ?

Далее для собранного макета спутника необходимо будет реализовать алгоритм стабилизации.

P. S.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru.

Практика II



4 ЧАСА

Алгоритм стабилизации

ЦЕЛЬ ЗАНЯТИЯ

Работа с собранной моделью спутника, разбор принципа стабилизации КА.

СТАБИЛИЗАЦИЯ КА

Активные системы стабилизации, о которых упоминалось ранее, имеют в своем составе три типа устройств:

- чувствительные элементы – устройства, собирающие информацию об окружающей среде (или иначе датчики); их функция заключается в сборе информации;
- исполнительные устройства – устройства, предназначенные для выполнения определенного действия, непосредственно влияющего на состояние всей системы (изменение положения, температуры и пр.);
- блок управления, который, опираясь на данные от чувствительных элементов (датчиков), формирует команды на исполнительное устройство.

В нашем случае, чувствительным элементом является датчик позиционирования (по-хорошему, он играет роль датчика угловой скорости, хотя, помимо угловой скорости, с него можно получать данные по направленности магнитного поля и линейной скорости), исполнительным устройством – двигатель-маховик, а блоком управления – плата Blue Pill.

Стабилизация спутника – это такое его состояние, в котором он поддерживает



4.17 ОБОБЩЕННАЯ СХЕМА РАБОТЫ СИСТЕМЫ СТАБИЛИЗАЦИИ И ОРИЕНТАЦИИ

постоянную заданную скорость вращения, компенсируя внешние воздействия. Это достигается путем создания алгоритма управления, который опирается на текущие данные об угловой скорости и задает такое управление на маховик (т. е. необходимую скорость вращения вала двигателя), которое позволяет удерживать космический аппарат в заданном состоянии.

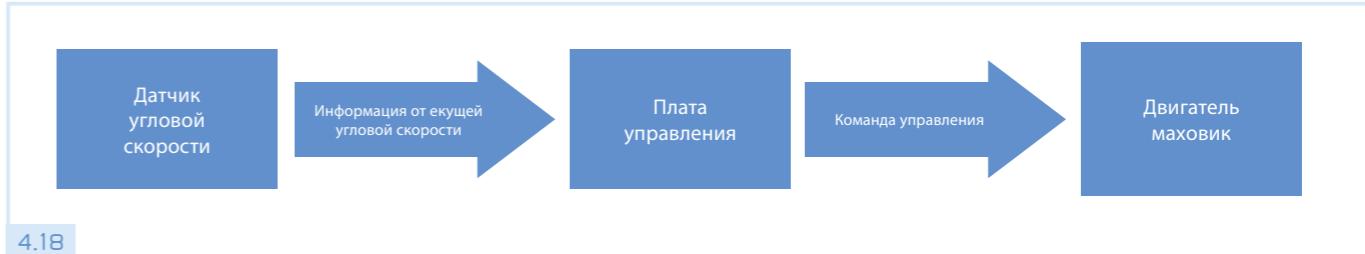
Для чтения показаний с датчиков набора IntroSat, работающих по I2C, используется библиотека «lS_Sensors». Она позволяет создать объект типа «Sensor» и использовать для него две функции:

`I2CwriteByte(uint8_t Address, uint8_t Register, uint8_t Nbytes, uint8_t* Data)` – позволяет передать на датчик байт информации.

Для данной функции необходимо в качестве входной информации указать:

`Address` – адрес slave-устройства;
`Register` – регистр slave-устройства, в который будет осуществляться запись данных;

4.18 ПОСЛЕДОВАТЕЛЬНОСТЬ СБОРА И ОБРАБОТКИ ИНФОРМАЦИИ



Nbytes – количество байт записываемых данных;
Data – указатель на массив данных, запись которых будет осуществляться.
I2Cread(uint8_t Address, uint8_t Register, int Size, uint8_t Data) – позволяет считывать показания с датчиков.

Для данной функции необходимо в качестве входной информации указать:

Address – адрес slave-устройства;
Register – регистр slave-устройства, с которого будет осуществляться чтение данных данных;
Size – количество байт считываемой информации;
Data – переменная, в которую будут записываться данные.

Пример кода для получения показаний угловой скорости:

```
#include <IS_Sensors.h>
#include <IS_Bluetooth.h>

#define ISMI_MPU_ADDR 0x68

#define GYRO_FULL_SCALE_250_DPS 0x00
#define GYRO_FULL_SCALE_500_DPS 0x08
#define GYRO_FULL_SCALE_1000_DPS 0x10
#define GYRO_FULL_SCALE_2000_DPS 0x18

//Задаем объект hyro
Sensor hyro;

void setup(){
Wire.begin();
Serial.begin(115200, SERIAL_8E1);

//Задаем конфигурацию датчика
hyro.I2CwriteByte(ISMI_MPU_ADDR, 26, 0x06);
hyro.I2CwriteByte(ISMI_MPU_ADDR, 27, GYRO_FULL_SCALE_1000_DPS);
}

void loop(){

if(Serial.available()){
char command=Serial.read();
if(command == 'b'){
enter_bootloader();
}
}

//Переменная для считывания данных
uint8_t Buf[6];

//Считываем 6 байт данных по шине I2C с
```

```
 регистра 0x43
hyro.I2Cread(ISMI_MPU_ADDR, 0x43, 6, Buf);

//Преобразуем считанные данные в
значения по осям
int16_t gx=-(Buf[0]<<8|Buf[1]);
int16_t gy=-(Buf[2]<<8|Buf[3]);
int16_t gz=Buf[4]<<8|Buf[5];

//Выводим данные по UART
Serial.print(gx, DEC);
Serial.print("\t");
Serial.print(gy, DEC);
Serial.print("\t");
Serial.print(gz, DEC);
Serial.print("\t");
Serial.println("");

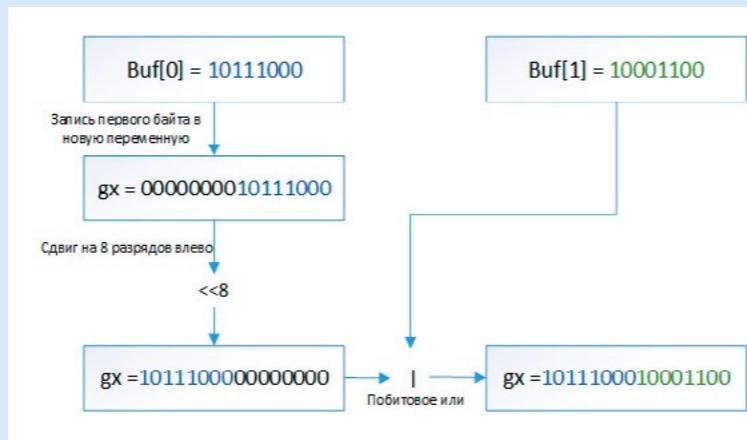
delay(500);
}
```

Строки с преобразованием

```
int16_t gx = -(Buf[0]<<8|Buf[1])
```

называются побитовым сдвигом. Данные угловой скорости по каждой оси выдаются датчиком в виде шестнадцатибитного числа или двух байт. Чтобы преобразовать два байта в одно число используется переменная типа *int16_t*, размер которой равен шестнадцати битам. *Buf[0]* и *Buf[1]* – элементы массива размером по 8 бит.

Таким образом, необходимо правильно соединить *Buf[0]* и *Buf[1]* в одной переменной.



Первый байт *Buf[0]* записывается по умолчанию в последние 8 разрядов, однако для правильной трактовки числа, его значение должно быть в первых восьми разрядах, если нумеровать разряды слева направо. Поэтому применяется оператор «`<<`», который сдвигает влево число на указанное количество знаков, а сдвинутые разряды заполняют нулями. После этого применяется оператор побитового ИЛИ «`|`», который выполняет операцию логического ИЛИ для каждого разряда.

Проверьте, что показания меняются при повороте кубсата.

Алгоритм для Arduino зациклен функцией *loop()*, поэтому определенный порядок действий, прописанный в ней, будет повторяться. Каждое такое повторение можно назвать итерацией. На каждой такой итерации должна задаваться скорость для вала двигателя. Примерная формула, по которой может вычисляться требуемая для режима стабилизации скорость вала двигателя, выглядит следующим образом:

$$\text{скор_max} = \text{стар_скор_max} - kd * (\text{угл_скор} - \text{треб_угл_скор})$$

где *скор_max* – значение скорости маховика, формирующееся на текущей итерации; *стар_скор_max* – значение скорости маховика на предыдущей итерации; *угл_скорость* – угловая скорость на текущей итерации; *треб_угл_скор* – требуемая угловая скорость; *kd* – коэффициент пропорциональности между скоростью маховика и ошибкой по угловой скорости.

В нашем случае рассматривается скорость вращения только вокруг вертикальной оси. Текущему положению датчика вертикальной оси соответствует ось Z, поэтому в формуле стоит учитывать только показания по этой оси.

Коэффициент пропорциональности задает соотношение между инерцией маховика и инерцией космического аппа-

ратов в целом. Такие коэффициенты достаточно сложно рассчитать из-за сложной формы космического аппарата, поэтому в нашем случае его значение легче всего подобрать опытным путем.

Таким образом, алгоритм для управления маховиком можно задать следующей блок-схемой – рис. 4.19.

Научившись управлять двигателем и снимать показания с датчика угловой скорости, не составит труда создать алгоритм стабилизации.

Учитывайте, что данные, получаемые с датчика угловой скорости, и управление двигателем изначально разного формата. Вы можете привести их к одному формату или поэкспериментировать с подбором коэффициента пропорциональности. Также учитывайте, что на вычисляемую скорость вала двигателя необходимо наложить ограничение, чтобы оно не превышало по модулю максимально возможное значение, равное 6000.

4.19 АЛГОРИТМ СТАБИЛИЗАЦИИ

ЧТО ДАЛЬШЕ?

Стабилизация является неотъемлемой частью более комплексной задачи: ориентации КА. В следующих модулях вас ждет знакомство со способами ориентации спутника в космическом пространстве, а также реализация одного из них на практике.

Р. С.

Если у вас остались вопросы, касающиеся проведения занятия – обязательно пишите их на почту info@introsat.ru



ЧАСТЬ V

ОРИЕНТАЦИЯ В КОСМИЧЕСКОМ ПРОСТРАНСТВЕ

ДАННЫЙ МОДУЛЬ ПОСВЯЩЕН СИСТЕМАМ ОРИЕНТАЦИИ КОСМИЧЕСКИХ АППАРАТОВ.

ОРИЕНТАЦИЯ КОСМИЧЕСКОГО АППАРАТА В ПРОСТРАНСТВЕ – ЭТО ПРОЦЕСС, КОТОРЫЙ ЗАДАЕТ АППАРАТУ ТАКОЕ ПОЛОЖЕНИЕ В ПРОСТРАНСТВЕ, КОТОРОЕ НЕОБХОДИМО ЕМУ ДЛЯ КОРРЕКТНОЙ РАБОТЫ. ЛОГИЧНО, ЧТО СОЛНЕЧНЫЕ БАТАРЕИ ДОЛЖНЫ БЫТЬ НАПРАВЛЕНЫ НА СОЛНЦЕ ДЛЯ ЭФФЕКТИВНОЙ РАБОТЫ, А КАМЕРА ДЗЗ – НА ЗЕМЛЮ. ПОСЛЕ ВЫВОДА НА ОРБИТУ, АППАРАТ ИМЕЕТ НЕКОТОРОЕ ВОЗМУЩЕННОЕ ДВИЖЕНИЕ, КОТОРОЕ ОН ДОЛЖЕН НОРМАЛИЗОВАТЬ И ОПРЕДЕЛИТЬ СВОЕ ПОЛОЖЕНИЕ В ПРОСТРАНСТВЕ.

НО ДЛЯ ТОГО ЧТОБЫ ЗАДАТЬ СЕБЕ ПРАВИЛЬНОЕ ПОЛОЖЕНИЕ, КОСМИЧЕСКИЙ АППАРАТ ДОЛЖЕН ДЛЯ НАЧАЛА ПОНЯТЬ, КАК ИМЕННО ОН РАСПОЛАГАЕТСЯ В ПРОСТРАНСТВЕ. СУЩЕСТВУЕТ НЕСКОЛЬКО СПОСОБОВ ЭТО СДЕЛАТЬ.

Теоретические основы



ГРУППОВАЯ
РАБОТА



2 ЧАСА

Способы ориентации КА

Первым шагом в ориентации является определение собственного положения в пространстве.

Положение в пространстве легче всего измерить, опираясь на какой-либо ориентир. На орбите у спутника есть несколько таких ориентиров:

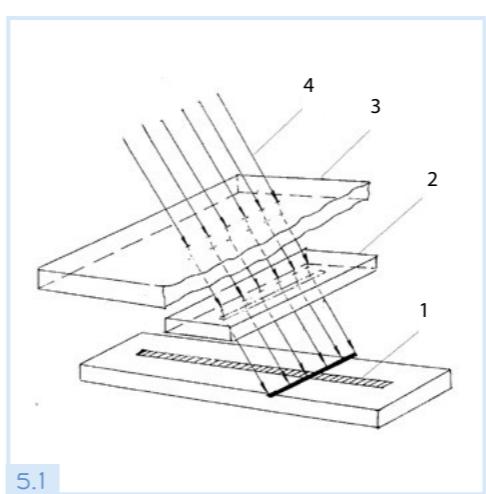
- Солнце, представляющее из себя яркий объект и достаточно однозначно задающее направление
- звезды
- Земля, как правило, располагающаяся ближе всего к космическому аппарату.

ОРИЕНТАЦИЯ ПО СОЛНЦУ

Зная, где находится Солнце, можно развернуть к нему космический аппарат соответствующей стороной. Как правило, для определения местоположения Солнца используют солнечные датчики.

Существует несколько видов солнечных датчиков, однако все они основаны на фоточувствительных элементах.

Часто в таких датчиках используются линейки ПЗС, реагирующие на падающий на них свет. Солнечные лучи проходят через щелевое отверстие и формируют



5.1



5.2

пятно на линейке фоточувствительных элементов. По характеру этого пятна можно определить угол падения солнечных лучей. Двух таких датчиков, расположенных под 90° относительно друг друга в одной плоскости, достаточно, чтобы определить направление, на котором расположено Солнце относительно космического аппарата.

Существуют и более сложные конструкции таких датчиков, имеющие одну щель в виде ломанной линии или пару разнесенных щелей. Тогда для определения направления на Солнце можно обойтись одним таким датчиком.

ОРИЕНТАЦИЯ ПО ЗЕМЛЕ

Существует несколько способов определения местоположения Земли.

ИНФРАКРАСНЫЙ ФОН

Так как Земля гораздо теплее, чем окружающее ее космическое пространство, по разнице температурного фона можно определить, в каком направлении планета расположена относительно спутника. Такие датчики имеют в своем составе инфракрасную оптическую систему и фокальную плоскость с термисторами.

Этот способ используется либо для высоких орбит, либо для низких. На высоких орbitах на фокальной плоскости формируется пятно от Земли, указывающее направление на нее. На низких орбитах используется принцип сканирования узким лучом определенной зоны, чтобы выявить расположение земного горизонта на фоне космического пространства.

МАГНИТНОЕ ПОЛЕ

Будучи на орбите, спутник все еще находится под действием магнитного поля Земли. Это позволяет определить положение планеты с помощью устройства, называемого магнитометром. Магнитометр определяет значение вектора магнитной индукции по трем осям. Полученное значение сравнивается с моделью магнитного поля, что позволяет определить положение спутника относительно Земли. Однако у этого способа есть недостатки: магнитометр чувствителен к возмущениям, что может приводить к неточной оценке.

5.1 УСТРОЙСТВО СОЛНЕЧНОГО ДАТЧИКА:

- 1) линейка фоточувствительных элементов;
- 2) щель;
- 3) светофильтр;
- 4) падающие солнечные лучи.

5.2 СКАНИРОВАНИЕ ИК-ГОРИЗОНТА ЗЕМЛИ

Практика I



ГРУППОВАЯ
РАБОТА



4 ЧАСА

Изучение ориентации по свету

ЦЕЛЬ ЗАНЯТИЯ

В ходе занятия изучается способ ориентации по солнечному свету с помощью четырех датчиков освещенности. По аналогии с реальными космическими аппаратами датчик света ставится на каждую грань.

ПОДГОТОВКА ПЛАТФОРМЫ

Процесс сборки платформы подробно описан в модуле «Исполнительные устройства». Однако в систему с маховиком и датчиком позиционирования добавляется еще четыре датчика освещенности.

В основе такого датчика лежит фотодиод, который меняет силу тока в зависимости от интенсивности падающего излучения, что и позволяет измерить величину падающего света.

Как и датчик позиционирования, датчик освещенности подключается к шине I2C, правда в отличие от него датчик освещенности может иметь до четырех адресов, что позволяет подключить к шине до четырех таких устройств. За адрес устройства отвечают переключатели на обратной стороне датчика.



5.3

Положение джампера на переключателе датчика	Адрес, соответствующий положению джампера
---	---



0x50



0x51



0x52



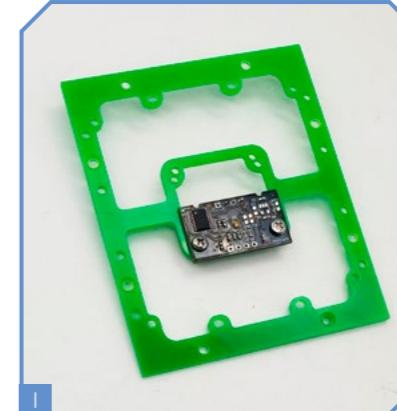
0x53

5.3 ДАТЧИК ОСВЕЩЕННОСТИ

ПОДГОТОВЬТЕ ДАТЧИКИ ОСВЕЩЕННОСТИ.

I Определите датчикам разные адреса, проверьте, чтобы не было датчиков с двумя одинаковыми адресами. С помощью двух винтов и гаек закрепите датчики на боковых панелях.

II Возьмите панель с датчиком, которому назначен I2C-адрес 0x50, на эту панель будет установлена солнечная батарея. Закрепите с обратной стороны панели пару металлических ответных пластин с резьбой при помощи винтов M2.5х6мм с потайной головкой, как показано на фото

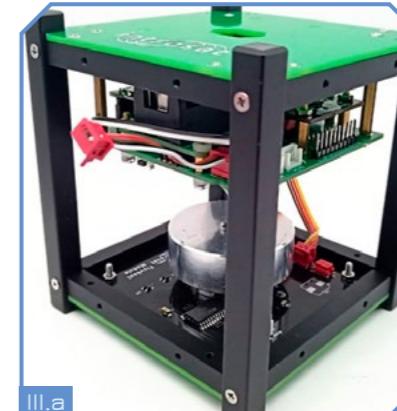


I



II

III.a-c Установите панели на раму при помощи винтов M2.5х6мм с потайной головкой. Подсоедините первый датчик к материнской плате с помощью шлейфа с разъемами MicroMatch. Следующие датчики можно последовательно подключать друг к другу по цепочке, используя вторые разъемы MicroMatch на каждом датчике.



III.a



III.b

IV С помощью пары специальных пластиковых уголков закрепите солнечную батарею на соответствующую грань как показано на фото. Контакты солнечной батареи можно подключить к разъему модуля питания, находящемуся с той же стороны, что и usb-разъем.



III.c

При сканировании шины I2C адреса всех устройств должны определяться.

```
Scanning...
I2C device found at address 0x33 !
I2C device found at address 0x50 !
I2C device found at address 0x51 !
I2C device found at address 0x52 !
I2C device found at address 0x53 !
I2C device found at address 0x68 !
done
```

5.4

ЧТЕНИЕ ДАННЫХ

Прежде всего попробуем считать данные об освещенности с датчиков по шине I2C. Для этого воспользуемся уже знакомой библиотекой «IS_Sensors».

Подключим необходимые библиотеки.

```
#include "IS_Sensors.h"
#include <IS_Bluetooth.h>
```

Зададим I2C-адреса датчиков.

```
#define ISMI_ADDR_1 0x50
#define ISMI_ADDR_2 0x51
#define ISMI_ADDR_3 0x52
#define ISMI_ADDR_4 0x53
```

5.4 ОПРЕДЕЛЕНИЕ I2C-АДРЕСОВ ВСЕХ УСТРОЙСТВ СИСТЕМЫ ОРИЕНТАЦИИ

Определим датчики освещенности.

```
SensorL1;
SensorL2;
SensorL3;
SensorL4;
```

Далее в функции setup инициализируем работу шины I2C и UART.

```
void setup() {
    Wire.begin();
    Serial.begin(115200, SERIAL_8E1);
}
```

В функции `loop` реализуем считывание данных.

```
void loop(){
    if(Serial.available()){
        char command=Serial.read();
        if(command == 'b'){
            enter_bootloader();
        }
    }

    uint8_t Buf[2];//переменная, в которую
    //считываются значения из I2C

    L1.I2Cread(ISMI_ADDR_1, 0, 2, Buf);//считываем значения с датчика L1
    int16_t light1 = (Buf[1] << 8 | Buf[0]);
    L2.I2Cread(ISMI_ADDR_2, 0, 2, Buf);//считываем значения с датчика L2
    int16_t light2 = (Buf[1] << 8 | Buf[0]);
    L3.I2Cread(ISMI_ADDR_3, 0, 2, Buf);//считываем значения с датчика L3
    int16_t light3 = (Buf[1] << 8 | Buf[0]);
    L4.I2Cread(ISMI_ADDR_4, 0, 2, Buf);//считываем значения с датчика L4
    int16_t light4 = (Buf[1] << 8 | Buf[0]);

    Serial.print (light1, DEC);
    Serial.print ("t");
    Serial.print (light2, DEC);
    Serial.print ("\t");
    Serial.print (light3, DEC);
    Serial.print ("\t");
    Serial.println (light4, DEC);
}
```

ОРИЕНТАЦИЯ ПО СВЕТУ

В разделе «Датчики на спутнике» разбиралась задача поворота солнечной батареи к максимальному источнику света. Алгоритм поворота был реализован достаточно просто, потому что шаговый двигатель жестко крепился в собираемой конструкции, а поворот на нужный угол задавался всего одной командой. Однако в состоянии невесомости такой способ реализовать невозможно из-за отсутствия опоры. По аналогии со стендом с солнечной батареей, макет кубсата также имеет четыре датчика, однако он находится в подвешенном состоянии, и именно в таком состоянии ему необходимо определить максимальный источник света и ориентироваться на него.

Алгоритм стабилизации немного похож на алгоритм ориентации: каждый из них работает с рассогласованием, т. е. ошибкой между желаемым значением параметра и

текущим. Однако если для стабилизации за регулируемый параметр бралась угловая скорость, то в случае ориентации этим параметром будет являться угол поворота кубсата относительно источника освещения. Обычно рассогласование обозначают буквой e :

$$e = x_{\text{текущее}} - x_{\text{идеальное}}$$

В данном случае x – регулируемый параметр. Как правило, текущее значение параметра x изменяется в зависимости от времени.

Как и в занятии по стабилизации, необходимо понять, каким именно способом нужно рассчитать требуемую скорость оборотов для маховика, чтобы рассогласование e стремилось уменьшаться к нулю.

Одним из широко используемых алгоритмов является, так называемый, способ ПИД-регулирования (пропорционально-интегрально-дифференциальное регулирование).

СОВЕТ. Можно показать учащимся небольшое видео о ПИД-регулировании: <https://www.youtube.com/watch?v=NbEhtZISa6A>

В данном способе регулирования есть три составляющих: пропорциональная, интегральная и дифференциальная. Каждая составляющая – это некоторая функция, зависящая от рассогласования. В случае пропорциональной составляющей – эта функция линейная, и ее можно записать как:

$$P = K_p \cdot e(t)$$

То есть эта составляющая меняется пропорционально текущему рассогласованию. В данном случае K_p – коэффициент, определяющий вес пропорциональной составляющей, т. е. насколько сильно в вычислениях необходимо учитывать данную часть.

Интегральная составляющая зависит от интеграла по рассогласованию:

$$I = K_i \int_{t_1}^{t_2} e(t) dt$$

K_i – коэффициент, задающий вес интегральной составляющей.

И последняя часть – дифференциальная –

это производная по рассогласованию:

$$D = K_d \frac{de(t)}{dt}$$

K_d – коэффициент, задающий вес дифференциальной составляющей.

И в общем случае скорость вращения маховика можно будет вычислить следующим образом:

$$\omega_{\text{wheel}} = P + I + D$$

СОВЕТ. Теория, связанная с ПИД-регулированием достаточно сложна, поэтому некоторые подробности в данном занятии опущены. Однако если учащиеся способны воспринять материал, то можно более подробно изучить общее устройство систем управления, принцип работы следящих систем и понятие переходного процесса.

Разберем пример реализации алгоритма, основанного на ПИД-регулировании. Для начала подключим все необходимые библиотеки:

```
#include <IS_Bluetooth.h>
#include <IS_Sensors.h>
#include <IS_Drive.h>
```

Определим адреса датчиков освещенности:

```
#define ISMI_ADDR_1 0x50
#define ISMI_ADDR_2 0x51
#define ISMI_ADDR_3 0x52
#define ISMI_ADDR_4 0x53
```

Зададим объекты датчиков и двигателя:

```
Sensor tri;
Sensor L1;
Sensor L2;
Sensor L3;
Sensor L4;
Drive motor;
```

Зададим переменные, которые будем далее использовать:

```
double sun_angle = 0;//угол нахождения
//максимального освещения
double angle_offset = 0;//переменная для
//расчета угла поворота кубсата
unsigned long int t, dt;//переменные для
//вычисления временных интервалов
unsigned long int t1, dt1;//переменные для
```

```
вычисления временных интервалов
double current_angle;//текущий угол
//поворота
double fixed_current_angle;
float e, p, i, d;//переменные, хранящие
//значения, необходимые для вычисления
//корректировки вращения двигателя
//маховика
//Коэффициенты пропорциональности
//для регулирования
float kr = 200;
float kp = 5;
float ki = 0.00002;
float kd = 5000000;
```

```
//Переменные для управления маховиком
short int return_dir;
long int return_speed;
```

```
//Переменные, использующиеся для
//хранения значений предыдущего цикла
//вычислений
short int z_speed_prev = 0;
unsigned long int t_prev = 0;
unsigned long int t_prev1 = 0;
float e_prev = 0;
```

Зададим в функции `setup` инициализацию необходимых интерфейсов:

```
void setup(){
    Wire.begin();
    Serial.begin(115200, SERIAL_8E1);
}
```

Далее по частям описывается функция `loop`:

```
void loop(){
    //Код для прошивки по bluetooth
    char command = Serial.read();
    if(command == 'b'){
        enter_bootloader();
    }
}
```

Добавим часть кода для считывания показаний с датчиков освещенности по I2C:

```
uint8_t Buf[2];//переменная, в которую
//считываются значения из I2C
```

```
L1.I2Cread(ISMI_ADDR_1, 0, 2, Buf);//считываем значения с датчика L1
int16_t light1 = (Buf[1] << 8 | Buf[0]);
L2.I2Cread(ISMI_ADDR_2, 0, 2, Buf);//считываем значения с датчика L2
int16_t light2 = (Buf[1] << 8 | Buf[0]);
L3.I2Cread(ISMI_ADDR_3, 0, 2, Buf);//считываем значения с датчика L3
int16_t light3 = (Buf[1] << 8 | Buf[0]);
L4.I2Cread(ISMI_ADDR_4, 0, 2, Buf);//
```

```
считываем значения с датчика L4
int16_t light4=(Buf[1]<<8|Buf[0]);
```

Далее упорядочим считанные данные с датчиков:

```
//Зададим массив с переменными,
содержащими показания датчиков
освещенности
```

```
int light[4]={
    light1,
    light2,
    light3,
    light4
};
```

```
//Отсортируем элементы массива в
порядке возрастания (от минимального
значения к максимальному)
std::sort(light, light + 4);
```

Определим сторону кубсата, на которую падает максимум света. Минимальное значение переменной датчика соответствует максимуму освещенности. После сортировки массива, переменная с минимальным значением станет первым элементом массива (т. е. `light[0]`). В соответствии со сборкой кубсата, чтобы повернуться датчиком с адресом 0x50 на источник света при максимуме на датчике с адресом 0x53, необходимо будет повернуться на угол 90°. Если максимум освещенности на датчике с адресом 0x51, то повернуться нужно будет на 90° в другую сторону, т. е. на -90°. Если же максимум освещенности находится на датчике с адресом 0x52, то нужно будет повернуться на 180°.

```
//Если максимум света на датчике с
адресом 0x53
if((light4 == light[0]) && (light[1] - light[0] >
200)) {
```

```
    sun_angle += 90;
```

```
}
```

```
//Если максимум света на датчике с
адресом 0x52
if((light3 == light[0]) && (light[1] - light[0] >
200)) {
```

```
    sun_angle += 180;
```

```
}
```

```
//Если максимум света на датчике с
адресом 0x51
if((light2 == light[0]) && (light[1] - light[0] >
200)) {
```

```
    sun_angle += -90;
```

```
//Если угол больше 360 градусов, то
делаем пересчет
sun_angle=sun_angle-((int(sun_angle)/
```

```
360)*360);
```

Выражение `light[1] - light[0] > 200` сравнивает между собой два самых минимальных значения. Если аппарат встает строго углом к источнику света, то значения на датчиках, расположенных на соседних гранях, будут не сильно различаться.

Далее считаем данные угловой скорости.

```
unsigned char data_array[6];
short int z_speed;
```

```
//Считываем показания угловой скоро-
сти по оси Z
```

```
tripl.I2Cread(0x68, 0x43, 6, data_array);
```

```
//Преобразуем показания угловой
```

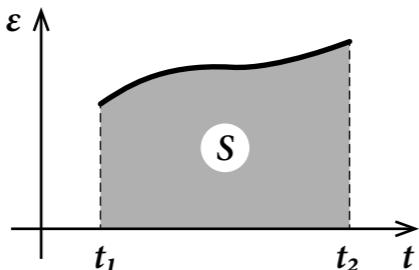
```
скорости по оси Z
```

```
z_speed = data_array[4]<<8|data_array[5];
```

Чтобы понимать, на какой угол необходимо повернуть аппарат, нужно знать угол, на какой угол аппарат повернулся. Из показаний угловой скорости можно вычислить угол поворота за некоторый момент времени. Так как угловая скорость – это производная от угла, то чтобы вычислить из угловой скорости именно угол необходимо проинтегрировать значение угловой скорости:

$$\alpha = \int_{t_1}^{t_2} \varepsilon(t) dt$$

Вспомним геометрический смысл интеграла. Значение интеграла равно площади фигуры под функцией, ограниченной слева и справа значениями аргумента. В нашем случае функцией является зависимость угловой скорости от времени $\varepsilon(t)$, а аргументом – время t .



Площадь фигуры под графиком определяется как площадь трапеции, т. е. полусумма оснований на высоту. В случае площади под графиком получается так, что трапеция как бы лежит на боку, и тогда основаниями будут являться значения угловой скорости $\varepsilon(t_1)$ и $\varepsilon(t_2)$, а высотой отрезок по оси абсцисс, соответствующий

временному отрезку $(t_2 - t_1)$. Тогда можно записать:

$$\alpha = \frac{\varepsilon(t_2) + \varepsilon(t_1)}{2} \cdot (t_2 - t_1)$$

Теперь напишем код, который вычисляет угол, на который поворачивается кубсат.

```
t=micros(); //текущий момент времени
dt=t-t_prev; //вычисляем длину про-
межутка времени
angle_offset=((double)(z_speed+z_speed_-
prev)/2)*(double)dt/130275454.272; //вы-
числяем угол поворота
z_speed_prev=z_speed; //запоминаем
текущее значение скорости как предыду-
щее для следующей итерации
t_prev=t; //запоминаем момент времени
как предыдущий для следующей итерации
```

Здесь деление на 130275454.272 переводит значения гироскопа в градусы.

Переменные `z_speed_prev` и `t_prev` изначально равны нулю.

```
current_angle+=angle_offset; //вычисляем,
на какой угол повернулся кубсат с нача-
ла своей работы
```

```
fixed_current_angle=current_angle-
((int(current_angle)/360)*360); //Если угол
больше 360 градусов, то делаем пересчет
```

Как говорилось ранее, весь процесс управления строится на определении текущего рассогласования. В нашем случае за рассогласование можно взять разность между текущим углом поворота кубсата и углом, где находится максимум освещенности.

```
//Определение рассогласования
e=sun_angle-fixed_current_angle;
```

Однако с исчислением углов не все просто. Например, если максимум освещенности будет находиться на датчике с адресом 0x51, то рассогласование будет равно 90°, и удобно повернуться в одну сторону. А если максимум освещенности на датчике с адресом 0x53, то рассогласование будет 270°, но удобнее повернуть на 90° в противоположную сторону, а не на 270° в ту же. Это можно учесть положительным или отрицательным значением рассогласования:

```
if(e>180){
    e=-360;
```

```
} else if(e<-180){
    e+=360;
}
```

Следующим шагом вычислим составляющие P , I и D . Для начала необходимо понимать, какой промежуток времени прошел, именно он задает в данном случае dt .

```
t1=micros(); //текущий момент времени
dt1=t1-t_prev; //вычисляем длину
промежутка времени
```

Переменная `t_prev` изначально равна нулю.

Вычислить пропорциональную составляющую не составляет труда:

```
p=kp*e; //определение пропорциональ-
ной составляющей
```

Как и в случае с вычислением угла поворота, используем геометрический смысл интеграла и вычислим интеграл как площадь под графиком функции. В данном случае функцией является изменение рассогласования $e(t)$, а аргументом все также пройденный промежуток времени:

```
i=ki*((e+e_prev)/2)*dt1; //определение
интегральной составляющей
```

Осталась последняя составляющая. В упрощенном виде можно рассматривать производную как отношение приращения функции к приращению аргумента:

$$\frac{de(t)}{dt} \approx \frac{\varepsilon(t_2) - \varepsilon(t_1)}{t_2 - t_1}$$

И тогда вычислить ее тоже не составит труда:

```
d=kd*(e-e_prev)/dt1; //определение
дифференциальной составляющей
```

```
e_prev=e; //запоминаем текущее
значение рассогласования как предыду-
щее для следующей итерации
t_prev1=t1;
```

```
return_speed=(int)(kr*(p+i+d)); //вы-
числяем требуемое значение скорости
маховика
```

Теперь осталось задать скорость врача-ния маховики.

```
//Определяем направление вращения
маховика
if(return_speed > 0){
    return_dir = 1;
} else {
    return_dir = 0;
}

//Ставим ограничение по допустимой
скорости вращения
if(return_speed > 5999){
    return_speed = 5999;
} else if(return_speed < -5999){
    return_speed = -5999;
}

//Задаем управление маховиком
motor.speed_control(0x33,return_dir,
abs(return_speed));

}
```

Полученный код необходимо загрузить на микроконтроллер и убедиться в том, что кубсат, размещенный на подвесе, будет поворачиваться к максимальному источнику освещения. В качестве источника освещения можно использовать фонарик на мобильном устройстве или настольную лампу.

В примере программы даны ориентировочные значения коэффициентов, поэтому изменяя их, можно добиться более точной работы алгоритма. Помимо этого, можно поэкспериментировать с самой логикой программы.

ЧТО ДАЛЬШЕ?

Датчик угловой скорости также снимает показания магнитного поля. Если его в достаточной мере изолировать от внешних помех, то можно попробовать осуществить ориентацию в магнитном поле. Для этого может быть полезным (хотя и не обязательным) использование колец Гельмгольца или любое иное оборудование, позволяющее создавать магнитное поле и управлять им.

Помимо ориентации по магнитному полю, можно также осуществить расчет векторов положения с помощью того же датчика угловой скорости. Используя данные с акселерометра, гироскопа и магнитометра, можно определить точное положение объекта относительно центра масс.

ДАЛЬНЕЙШИЕ РЕКОМЕНДУЕМЫЕ МОДУЛИ:

- Орбитальная механика
(Почему спутники не падают)
- Механические конструкции
- Обработка космических снимков

ПРИЛОЖЕНИЕ 1

СПЕЦИФИКАЦИЯ НАБОРА

Компоненты для вводного практикума

Переменный резистор  1 шт.

Светодиод  5 шт.

Резистор 220 Ом  5 шт.

Макетная плата  2 шт.

Корпусные элементы

Верхняя панель  1 шт.

Нижняя панель  1 шт.

Боковая панель  4 шт.

Рельсы  4 шт.

Перекладина с пазом  4 шт.

Перекладина с торцевым отверстием  4 шт.

Маховик  1 шт.

Основные платы и компоненты

Плата Blue Pill  1 шт.

Материнская плата  1 шт.

Модуль питания  1 шт.

Модуль интерфейса и прототипирования  1 шт.

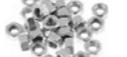
Модуль маховика  1 шт.

Датчик освещенности  4 шт.

Датчик позиционирования  1 шт.

Шаговый двигатель в корпусе  1 шт.

Служебные модули и вспомогательные компоненты	
Программатор ST-Link V2	 1 шт.
Конвертер usb-uart (внешний вид может отличаться)	 1 шт.
Bluetooth-модуль	 2 шт.
Аккумулятор Li-Ion 18650, защищенный	 1 шт.
Цифровой амперметр (Внешний вид может отличаться)	 1 шт.
Солнечная батарея с комплектом крепежных элементов	 1 шт.

Крепеж	
Винты M2.5x4 с полукруглой головкой	 10 шт.
Винты M2.5x6 с полукруглой головкой	 50 шт.
Винты M2.5x12 с полукруглой головкой	 25 шт.
Винты M2.5x6 с потайной головкой	 50 шт.
Винты M2.5x12 с потайной головкой	 25 шт.
Латунные стойки M2.5x5	 6 шт.
Латунные стойки M2.5x11	 25 шт.
Гайки M2.5	 40 шт.
Шайбы изоляционные M2.5	 10 шт.

Подвес	
Рейка, 30 см	 4 шт.
Рейка, 15 см	 2 шт.
Карабин	 1 шт.
Вертлюг с подвесным элементом	 1 шт.
Соединитель, куб	 2 шт.

Соединитель, внутренний угол	 6 шт.
Шестигранный ключ 2.5	 1 шт.
Шестигранный ключ 4 мм	 1 шт.
Пластина для крепления шагового двигателя	 1 шт.
Пазовый сухарь	 1 шт.

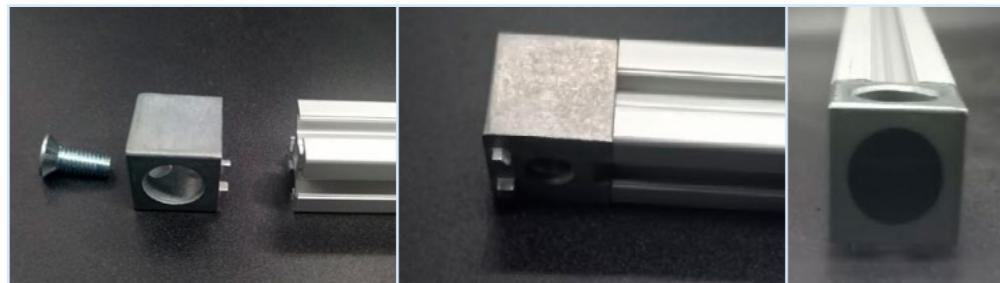
Монтажные провода и кабели	
Беспаячные провода «папа-папа»	 20 шт.
Беспаячные провода «мама-мама»	 20 шт.
Беспаячные провода «папа-мама»	 20 шт.
Шлейф AWG28 8 жил, 1 метр	 1 шт.
Разъемы MicroMatch на шлейф	 20 шт.
Провод питания	 2 шт.
Кабель usb – microusb	 2 шт.

ПРИЛОЖЕНИЕ 2

ИНСТРУКЦИЯ ПО СБОРКЕ ПОДВЕСА

76

Четыре рейки длиной 30 см отличаются друг от друга по наличию резьбы. У верхней с каждого торца есть резьба. У боковых реек только с одной стороны есть резьба. У нижней резьбы нет.



■ На боковые рейки длиной 30 см установите с помощью винтов M6 кубические соединители, используя для этого шестигранник 4 мм. На верхнее отверстие кубического соединения установите заглушку.



■ На верхнюю рейку установите в паз крепление для карабина.



■ Соедините боковые рейки с верхней рейкой, закрепив кубические соединения винтами. Затем на боковые отверстия кубических соединителей установите заглушки.



■ В паз нижней рейки установите пазовый сухарь.



■ Установите на нижнюю рейку угловые соединители с каждого конца.



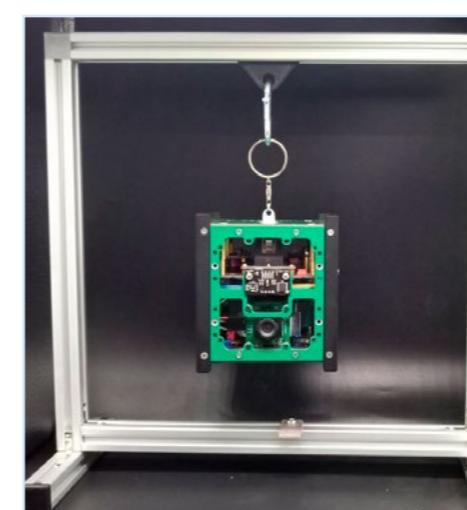
■ Установите собранную из трех реек часть на нижнюю рейку так, чтобы угловые соединители, установленные на нижней рейке встали в пазы боковых реек. Закрепите угловые соединители с помощью шестигранника.



■ Установите на каждую рейку длиной 15 см по два угловых соединителя. По бокам установите заглушки.



■ Возмите две рейки 15см, это опоры подвеса. Вставьте угловые соединители в пазы вертикальных стоек и затяните при помощи шестигранника 2.5 мм



■ На сухарь установите пластину с двумя отверстиями и закрепите винтом.

■ 7. Подвес готов.

77

ПРИЛОЖЕНИЕ З ПРОШИВКА МИКРОКОНТРОЛЛЕРА ЧЕРЕЗ BLUETOOTH-МОДУЛЬ

В комплектацию конструктора IntroSat входят два bluetooth-модуля, с помощью которых можно настроить беспроводную прошивку конструктора. Помимо прошивки bluetooth-соединение также позволяет считывать данные по беспроводной реализации Serial-порта. Это гораздо удобнее, чем подключение ST-Link и конвертера usb-uart для реализации тех же функций. Однако данный способ требует преднастройки, которая пошагово приведена ниже.

1. Перевод bluetooth-модуля в AT режим для настройки.

К конвертеру usb-uart подключаем один из bluetooth-модулей. Пины STATE и EN на модуле остаются не подключенными. Теперь нужно перевести модуль в AT режим. При отключенном от ПК конвертере usb-uart зажимаем кнопку на модуле (она на нем одна) и подключаем usb-uart к ПК, кнопку отпускаем. Если светодиод на модуле стал медленно мигать, то все правильно.

2. Проверка связи и настройка монитора порта в Arduino IDE.

Запускаем Arduino IDE, выбираем порт, открываем монитор порта (под Ubuntu порт закрывается только если запустить IDE через терминал командой «sudo arduino»).

Ставим настройки:



Пишем в строку «AT». На первую отправку, скорее всего, в порт выводится «ERROR». Необходимо повторно отправить в порт «AT»; как правило, после этого выводится «OK». Это значит, что модуль готов к настройке, можно переходить к пункту 3.



Если в монитор порта ничего не выводится, скорее всего перепутано, подключение rx и tx. Попробуйте поменять провода местами.

ВАЖНО! Если светодиод на bluetooth-модуле начинает быстро мигать после ввода AT-команды (как при поиске другого модуля), значит он перезагрузился и вышел из AT-режима. В таком случае, чтобы продолжить настройку нужно снова выполнить пункты 1 и 2, далее можно вводить следующую команду.

3. Настройка ведомого модуля через Arduino IDE.

ВАЖНО! Все команды в строке монитора порта необходимо прописывать заглавными буквами.

Пишем последовательно через монитор порта следующие команды:

1) AT+UART=115200,0,2

Команда задает настройку uart-соединения (нам пока подходят только такие параметры, потому что они должны совпадать с параметрами stm32cubeprogrammer, который загружает сам код на STM).

```
Serial Port ttyUSB0 is successfully opened.
Port configuration: parity = even, baudrate = 115200, data-bit = 8,
stop-bit = 1,0, flow-control = off
Activating device: OK
```

2) AT+ROLE=0

Назначаем модуль ведомым. Ведомый модуль будет подключаться к плате Blue Pill, он является бортовым.

3) AT+ADDR?

Спрашиваем mac-адрес модуля и обязательно записываем его, чтобы не потерять; он необходим для последующих настроек.

```
ERROR: (0)
OK
OK
OK
OK
+ADDR:98d3:b1:fd3973
OK
```

4. Пересменка.

Отключаем ведомый модуль от ПК. Для удобства стоит его подписать, чтобы не перепутать ведущий и ведомый модули. Вместо ведомого подключаем второй модуль; он будет у нас ведущим. Его также надо перевести в AT режим. Далее, также как с первым модулем, переходим в Arduino IDE, проверяем связь (п. 2). Теперь можно перейти к п. 5.

5. Настройка ведущего модуля через Arduino IDE.

Пишем последовательно через монитор порта следующие команды:

1) AT+UART=115200,0,2

2) AT+ROLE=1
Задаем модулю роль ведущего.

3) AT+CMODE=1
Разрешаем подключаться только к модулю, к которому он привязан.

4) AT+BIND={MAC}

Привязываем модуль к mac-адресу ведомого модуля (двоеточия заменяются запятыми). Для определенного в п.3 адреса команда выглядит так: AT+BIND=98d3,b1,fd3973.

При ошибке «ERROR:(7)», обратите внимание на MAC-адрес bluetooth-модуля. Он выводится в следующем формате: XXXX,XX,XXXXXX. Если в ответе от ведомого модуля на команду «AT+ADDR?» содержался MAC адрес, не соответствующий формату, указанному выше (например, 35:c1:45fd), значит вам при вводе MAC-адреса для ведущего модуля нужно добавить нули перед группами символов MAC-адреса (например, «AT+BIND=0035,c1,0045fd»). Это происходит не на всех модулях, зависит от версии прошивки конкретного модуля.

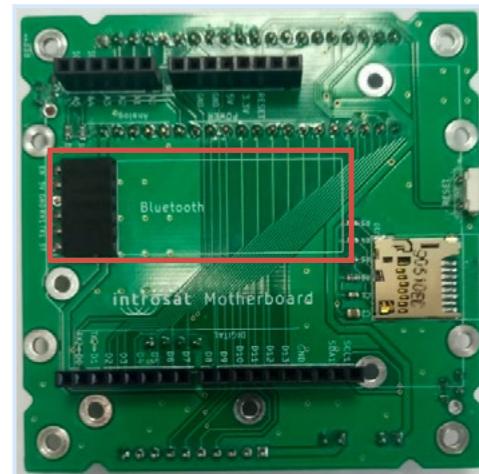
Также обратите внимание на то, есть ли в конце вводимой строки с адресом пробел – его быть не должно. Наличие пробела также может вызывать «ERROR:(7)».

Привязываем модуль к mac-адресу ведомого модуля (двоеточия заменяются запятыми).

Для определенного в пункте 3 адреса команда выглядит так: AT+BIND=98d3,b1,fd3973. Теперь нужно переподключить ведущий модуль, чтобы он вышел из AT режима.

6. Подключение к STM.

Подключаем ведомый модуль к STM32: для этого на материнской плате есть специальное гнездо.



7. Перевод STM в режим bootloader'a.

Зальем через ST-Link код, приведенный ниже (в Arduino IDE необходимо поставить Upload Method: SWD).

Библиотеку «IS_Bluetooth» можно скачать по ссылке: introsat.ru/download.

```
#include <IS_Bluetooth.h>

void setup() {
  Serial.begin(115200, SERIAL_8E1);
}

void loop() {
  char msg = Serial.read();
  if(msg == 'b') {
    enter_bootloader();
  }
}
```

Все что делает этот код, это вызывает функцию, переводящую STM в режим бутлоадера, при отправки в монитор порта буквы «б».

ВАЖНО! Обязательно проверьте, чтобы джамперы на текущий момент находились в исходном положении (оба в положении 0).

Убедитесь, что оба Bluetooth модуля быстро мигают 2 раза (каждые 2 секунды); это означает, что они подключены друг к другу.

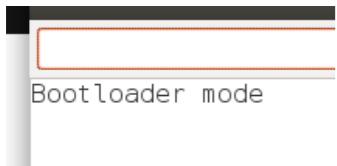
8. Загрузка прошивки

Чтобы загрузить любой скетч, выбираем в Arduino IDE во вкладке «Инструменты» Upload Method - Serial.

Далее заходим в монитор порта и ставим параметры:



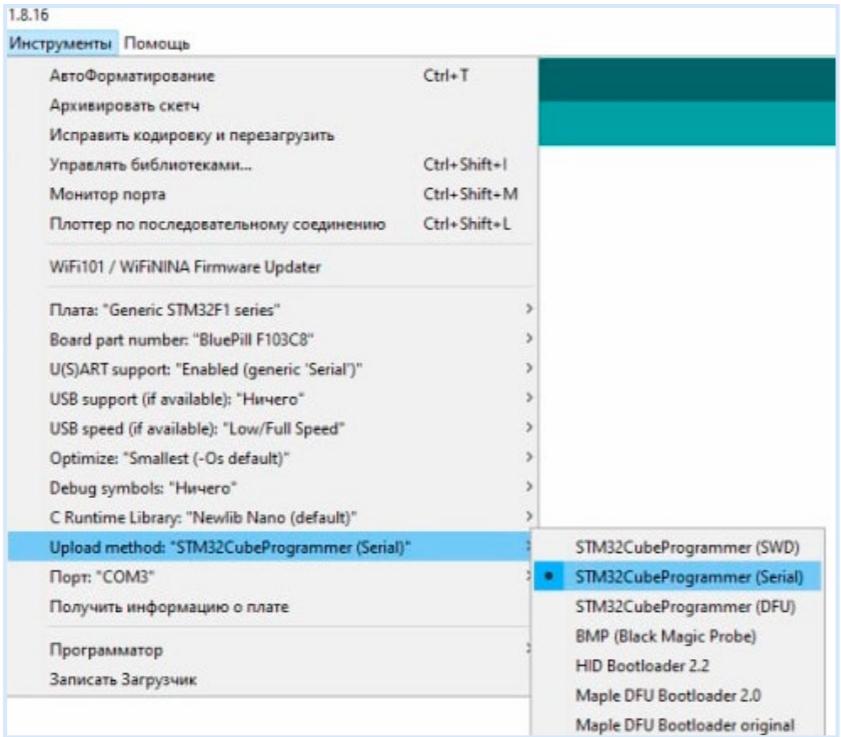
Чтобы STM перешла в режим загрузчика, отправляем впорт символ «б» и ждем появления сообщения «Bootloader mode».



Жмем кнопку: 

Если все правильно, то программа загружается на STM удаленно.

Чтобы загружать на STM скетчи удаленно, необходимо будет всегда добавлять в них приведенный выше код. Однако в большей части примеров этот код уже содержится.



СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Бертяев В. Д. Теоретическая механика (краткий конспект лекций). Учеб. пособие // В.Д. Бертяев, Л.А. Булатов, С.С. Маркелов; Тул. гос. ун-т, Тула, 2005. – 144 с.

Демаков Ю. П. Лекции по физическим основам электроники: Учебное пособие для вузов. Ижевск: Изд-во ИжГТУ, 2008. – 323 с.: ил.

Фортескью П. Разработка систем космических аппаратов // Фортескью П., Сайрнерд Г. Старк. Д.; пер. с англ. М.: Альпина Паблишер, 2015. – 765 с.

Гагарин Александр. Arduino для начинающих. Урок 3. Подключение потенциометра. // Занимательная робототехника. 2014. – URL <http://edurobots.ru/2014/04/arduino-potenciometr/> (дата обращения: 01.08.2019).

Клименко К. А. Сравнительный анализ современных датчиков тока // Молодой ученый. 2011. №8. Т.1. – 66-68 с. – URL <http://moluch.ru/archive/31/3552/> (дата обращения: 01.08.2019).

Павлов Андрей. I2C-адрес устройства и как его узнать // Вольтик. 2017. – URL <http://voltiq.ru/how-to-find-a-device-i2c-address/> (дата обращения: 01.08.2019)

Сабуров П. С. Учебное пособие к курсу лекций по дисциплине «Автоматическое управление, датчики и приборы»: [в ч.] Ч. 1 // Электронная библиотека ВлГУ. 2016. – URL <http://e.lib.vlsu.ru:80/handle/123456789/3685> (дата обращения: 01.08.2019)

Сальбах Артур. MPU 9250 и Arduino – схема подключения. // Вольтик. 2017. – URL <http://voltiq.ru/mpu-9250-and-arduino/> (дата обращения: 01.08.2019)

Шаговые двигатели, принцип действия, разновидности, характеристики // My Practic. 2017. – URL <http://mypractic.ru/shagovye-dvigateli-princip-dejstviya-raznovidnosti-karakteristiki.html> (дата обращения: 01.08.2019)

МЕТОДИЧЕСКИЕ ПОСОБИЯ КУРСА

1. Ракетостроение
2. Датчики на спутнике
3. Исполнительные устройства
4. Ориентация космических аппаратов
5. Электробаланс
6. Теплообмен
7. Механические конструкции
8. Радиосвязь
9. Космическая оптика
10. Научные эксперименты в космосе
11. Орбитальная механика
12. Орбитальное маневрирование
13. Прием космических данных
14. Обработка космических снимков

82

ДЛЯ ЗАМЕТОК

83

ДЛЯ ЗАМЕТОК

84



Современный подход к построению
Урока Технологии и проектного обучения



КУРСЫ ДЛЯ ПОДГОТОВКИ НАСТАВНИКОВ

Экспресс-курсы по школьной проектной деятельности в области космонавтики и по работе с оборудованием, программа повышения квалификации.



МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ И НАБОРЫ ДЛЯ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ И УРОКОВ ТЕХНОЛОГИИ

Школьный курс космонавтики для учащихся 8-11 классов. Модули по 6-12 часов раскрывают различные аспекты космической инженерии и связанные с ними темы физики, информатики и робототехники.



НАГЛЯДНАЯ АСТРОНОМИЯ НА МЕСТЕ ОБУЧЕНИЯ

Комплекс планетария, устанавливающегося в условиях обычного прямоугольного помещения, включающий специализированное ПО и методические материалы для расширенного курса астрономии.



ПРАКТИКУМЫ ПО КОСМОНАВТИКЕ ДЛЯ СТАРШИХ КЛАССОВ И ПОДГОТОВКА К УЧАСТИЮ В СОРЕВНОВАНИЯХ

Учебные конструкторы и комплексы лабораторного оборудования, разработанные для введения в практику работы с микроспутниковыми платформами и другими аспектами космонавтики, в том числе применяющиеся в компетенции «Инженерия космических систем» WorldSkills, этапах космических профилей Олимпиады НТИ и соревнований КанСат.



ПОГРУЖЕНИЕ В ЗАДАЧИ НЕБЕСНОЙ МЕХАНИКИ В СИМУЛЯТОРЕ КОСМИЧЕСКИХ ПОЛЕТОВ

Специализированное программное обеспечение, позволяющее:

- Вести отдельный курс без расходных материалов;
- Расширять и обогащать физикой другие курсы;
- Готовиться к соревнованиям в области космонавтики;
- Проводить собственные инженерные соревнования.



РАБОТА С ПРОФЕССИОНАЛЬНЫМИ ИНСТРУМЕНТАМИ МОНИТОРИНГА ЗЕМНОЙ ПОВЕРХНОСТИ

Стационарные центры приема спутниковых данных и управления, позволяющие школьникам и студентам не только знакомиться с задачами наземного сегмента космической индустрии, но и принимать участие в решении актуальных задач экологии и экономики.



ПРОЕКТЫ НА БАЗЕ ПРОФЕССИОНАЛЬНЫХ ПЛАТФОРМ КОСМИЧЕСКИХ АППАРАТОВ

Коммерческие нано- и микро-спутниковые платформы, позволяющие работать как с учебными, так и летными образцами космических аппаратов, оборудование предполетной подготовки и консультационная поддержка.



8 800 550-0492

www.introsat.ru

info@introsat.ru

УДК 629.783
ББК 39.64
Я499



По вопросам технической и методической поддержки, обучения и поставок:

info@introsat.ru

8 800 550-0492

www.introsat.ru

