

Проект UltraBird

Blokhin Yuri
ultrablox@gmail.com

13 іюля 2015 г.

Оглавление

1	Концептуальное описание	1
1.1	Основная цель проекта	1
1.2	Используемые инструменты	1
2	Решаемые задачи	3
2.1	Общая информация	3
2.2	Основные задачи	3
2.2.1	Перемещение в заданные координаты	3
2.2.2	Сопровождение	3
2.3	Дополнительные задачи	4
2.3.1	Перенос полезного груза	4
2.3.2	Прямое наблюдение	4
2.3.3	Фото-/видео-разведка	4
2.3.4	Геодезия	4
2.3.5	Агрессивное маневрирование	5
3	Структура проекта	6
3.1	Аппаратная декомпозиция	6
3.1.1	Переносимая часть	6
3.1.2	Стационарная часть	7
3.1.3	Носитель	7
4	План разработки	8
4.1	Фаза 1	8
5	Ссылки на похожие проекты	9
6	Сравнение с конкурентами	10

Аннотация

Данный документ дает детальное описание проекта UltraBird. В нем дано концептуальное описание, декомпозиция проекта на небольшие подзадачи, а также список задач, решаемых с помощью разрабатываемого программно-аппаратного комплекса.

Глава 1

Концептуальное описание

1.1 Основная цель проекта

Цель заключается в разработке система автопилота (СА) для узкого класса автономных аппаратов, а именно: небольшие (до 3-х метров по наибольшему измерению) модели автомобилей, вертолетов и иных летательных аппаратов. СА представляет собой программно-аппаратный комплекс, состоящий из нескольких аппаратных модулей, соединяемых посредством общего интерфейса. В системе присутствует центральный модуль, в котором сосредоточена вся высокоуровневая логика. Данный модуль находится под управлением системы реального времени (СРВ).

СА рассчитана на поддержание связи с базой, и получения полетных заданий. Полетное задание определяет текущее полетное задание, которое подразумевает автономное выполнение. Например, полетным заданием может являться: переместиться в координаты (X,Y), минуя препятствия.

Система может быть использована для управления различными платформами (далее носителями), при этом, за счет унификации интерфейсов управления носителем, переконфигурация СА будет производиться сменой прошивки/настройки.

1.2 Используемые инструменты

В качестве основного языка проекта для написания основного ПО выступают C/C++, компилируемый под целевую СРВ. Отсюда возможны ограничения на возможности последних версий языка, и, вероятно, определенные ограничения в поддержке шаблонов.

При написании ПО для контроллеров предпочтительно использование C/C++, однако в особых случаях допускается использование ассемблера, при наличии на то серьезных причин.

В качестве СРВ используется операционная система (ОС) QNX. Несмотря на разнообразие поддерживаемых данной ОС плат, на данный момент

выбрана плата BeagleBone для использования в качестве центрального модуля.

Глава 2

Решаемые задачи

2.1 Общая информация

Часть описанных задач опирается на построение модели окружающего мира, на основе информации поступающей с камер и датчиков расстояния. В зависимости от технических возможностей, которых реально можно достичь, некоторые задачи могут быть изменены или убраны. Здесь также приветствуются любые разумные идеи.

Стоит отметить, что большая часть задач была придумана исходя из того, что носителем будет мультикоптер. Поэтому далеко не все задачи, решаемые на одном носителе могут быть решены на другом носителе.

В качестве первой фазы проекта рассматривается наземный колесный носитель, исходя из этого в приоритет выносятся соответствующие ему задачи.

2.2 Основные задачи

Здесь описываются задачи, от решение которых является критическим при выполнении проекта.

2.2.1 Перемещение в заданные координаты

Данная задача является первичной, и заключается в возможности перемещения в заданные координаты с объездом препятствий. При этом, грубо маршрут можно задать несколькими контрольными точками, которые необходимо посетить.

2.2.2 Сопровождение

Носитель должен иметь возможность сопровождать выбранный оператором объект. Сопровождение и выбор объекта подразумевают нахождение

объекта в фокусе камеры. Как и в предыдущем пункте, подразумевается возможность выполнения этой задачи вне зоны связи с терминалом. Также подразумевается, что время, отведённое на выполнение задачи, явно (оператором) или неявно (зарядом батарей) ограничено.

2.3 Дополнительные задачи

2.3.1 Перенос полезного груза

БПЛА должен иметь возможность переносить полезный груз, доставлять его в заданную точку. При этом на аппарате закрепляется груз и ему сообщается траектория полета, либо конечная точка (тогда БПЛА должен пытается проложить траекторию самостоятельно). Так же в пределах данной задачи аппарат должен летать и без груза (с нулевым полезным грузом).

БПЛА должен иметь управляемый захват для груза. При этом, захват должен контролироваться как с управляющей станции, так и самим БПЛА. Должна быть возможность сбрасывать груз таким образом, чтобы он падал максимально близко к выбранной точке. При этом БПЛА оперирует следующими данными: координаты точки сброса, снимок местности.

2.3.2 Прямое наблюдение

БПЛА должен быть способен выполнять роль наблюдателя. Под наблюдением подразумевается, что БПЛА напрямую выполняет команды оператора по примитивам перемещения, и предоставляет оператору изображения с бортовых видеокамер.

2.3.3 Фото-/видео-разведка

БПЛА должен иметь возможность совершать автономный полет, в том числе за границей радиосвязи с терминалом. Пройденная траектория должна анализироваться с тем, чтобы аппарат мог вернуться в зону связи. Полученная информация в случае невозможности немедленной передачи на терминал должна сохраняться на бортовом накопителе, а после входа в зону связи по запросу передаваться на терминал.

2.3.4 Геодезия

БПЛА должен указывать при помощи лазера (или иного источника света) точки на поверхности. Точность указания должна соответствовать точности реальных геодезических приборов. Задача подразумевает постоянную обратную связь с оператором.

2.3.5 Агрессивное маневрирование

БПЛА должен уметь отрабатывать сложную пространственную траекторию в пределах заданной погрешности. Промежуточные положения задаваемой траектории могут быть неустойчивыми для аппарата.

Глава 3

Структура проекта

3.1 Аппаратная декомпозиция

Аппаратно проект состоит из трех больших групп - стационарная станция, с которой происходит управление, мозг самого летательного аппарата, а также платформа-носитель. Наиболее сложной с точки зрения разработки частью является мозг ЛА.

3.1.1 Переносимая часть

Модуль шасси отвечает за управление носителем. Как правило, все управление сводится к множеству выходных ШИМ-сигналов, воспринимаемых стандартным RC-оборудованием. Также, скорее всего потребуется дополнительные выходы (в идеале - через тот же ШИМ с теми же параметрами) для управления дополнительным навесным оборудованием (например, свет, лазер, манипулятор для захвата груза). В зависимости от носителя, возможно наличие нескольких модулей шасси в одном аппарате (в случае нехватки каналов).

Вычислительный модуль отвечает за исполнение вычислений и обеспечивать работу всего БПЛА в целом. Его наличие обязательно. Сохраняет данные о полете (траектория, показания датчиков) как черный ящик на SD-карту (возможно дублирование). Управляется ОС реального времени. Также решает задачу стабилизации ЛА.

Сенсорный модуль отвечает за ориентацию БПЛА в пространстве. Имеет несколько датчиков, которые регулярно опрашивает, а также интегрирует траекторию.

Радиомодуль отвечает за связь с управляющей станцией. Расчитан на передачу малого битрейта управляющей информации, а также на передачу потокового видео (по разным каналам).

Силовой модуль должен отвечать за управление винтовыми двигателями. Так же должен отвечать за посадку аппарата в случае обнаружения

серьезных неполадок в бортовом оборудовании (частичный либо полный выход из строя любого другого обязательного модуля).

Видеомодуль должен отвечать за снятие сигнала с видеокамер, кодирование и сжатие сигнала и отправку сигнала на радиомодуль. Возможно, для снижения нагрузки на общие шины, можно интегрировать с радиоканалом для радио.

3.1.2 Стационарная часть

Представляет из себя устройство, подключаемое к компьютеру, и имеющее радиоканал связи с мозгом БПЛА. Позволяет отправлять управляющие команды на БПЛА, запрашивать с него различную информацию, а также получать видео в реальном времени. Имеет платформо-независимый интерфейс с компьютером через какой-либо стандартный интерфейс.

3.1.3 Носитель

Может представлять из себя автомобиль/вертолет/мультикомптер/самолет, либо что-то другое. Единственным необходимым условием является полная управляемость с помощью оборудования, управляемым в свою очередь ШИМ-сигналом.

Глава 4

План разработки

4.1 Фаза 1

Производится разработка базовых модулей на макетной плате, а именно: модуля шасси и навигационного модуля. Производится написание программного обеспечения на контроллеры модулей. Первым делом производится разработка модуля шасси.

Производится макетирование модуля шасси.

Производится реализация ПО главного модуля для управления модулем шасси.

Далее производится выполнения модуля шасси в железе.

Далее производится макетирование навигационного модуля, выбор датчиков и написание ПО.

Производится написание ПО главного модуля для работы с навигационным модулем.

Далее производится выполнение навигационного модуля в железе.

Глава 5

Ссылки на похожие проекты

Описание проблем, возникших при создании автопилота для коптера <http://www.tdxcopter.com/projects/updates>

Сенсорная плата для BeagleBone <http://michaelhleonard.com/projects/sensorcape/>
<https://github.com/LeonardMH/SensorCape>

Глава 6

Сравнение с конкурентами

Разница в том, что существует много автопилотов для коптеров, которые позволяют просто вести упрощенный полет с помощью классической радио-аппаратуры. а в моем варианте это будет устройство, которое можно будет подключить к любому носителю (будь то машинка или танк, или вертолетик), загрузить нужную программу управления, и оно поедет.

Литература

- [1] Leslie Lamport, \LaTeX : A Document Preparation System. Addison Wesley, Massachusetts, 2nd Edition, 1994.