Описание ПО БК

# Структура проекта

Проект представлен в виде нескольких директорий, являющихся папками проектов с/с++. Описание директорий проекта.

camera - приложение, производящее фото съемку.

lib - основные библиотеки проекта и приложение для их тестирования

kk – приложение управления критическим контроллером

barometer – приложение для снятия показаний с барометра

termo – приложение для снятия показаний датчиков температуры (не реализовано)

memsNavigation – приложение для управления акселерометром, гироскопом, магнитометром (не реализовано)

gpsNavigation – приложение для управления навигационным приемником (не реализовано)

tr – приложение управления приемопередатчиком (не реализовано)

# Автозапуск

Запуск приложений при старте БК осуществляется с помощью системы скриптов. При старте БК исполняется системный скрипт /etc/rc.local в который добавлен скрипт проекта. На данный момент скрипт проекта должен иметь путь и название “/home/pi/rcs/test\_run.sh”. В нем приложения необходимо запускать в параллельном режиме. Например “camera &”. Символ & означает, что скрипт будет запускать приложение в подпроцессе и не будет блокировать свое исполнение.

# Хранилища телеметрии.

Для работы с телеметрией используются специальные хранилища данных. Для доступа к этим хранилищам необходимо использовать наследников класса dataService. В них реализован механизм добавления телеметрии по ее типу. Во всех классах dataService реализованы следующие методы:

connect() – служит для подключения к сервису

disconnect() – служит для отключения от сервиса

add(int type, char\* data, int dataSize) – служит для добавления информации в хранилище. При сохранении добавляет временную метку данным для последующего доступа к ним. Здесь type – тип телеметрии, data- указатель на телеметрию, dataSize – размер телеметрии в байтах

В остальном классы не стандартизированы.

Существует три реализации класса dataService.

1. Реализован с использованием семафоров и разделяемой памяти. Класс называется «IPCDataService». И может быть проинициализирован со следующими параметрами: serviceName – имя сервиса, используемое для подключения к сервису, memSize – размер памяти, выделяемый для хранения информации сервиса, memType – тип памяти с возможными значениями MEM\_SHARED для работы с оперативной памятью и MEM\_FILE для работы с энергонезависимой памятью.
2. Реализован с помощью простой linux БД – «DBM» . Класс называется «DBMDataService» и позволяет писать и читать данные базы по ключу.
3. Реализован с помощью БД «Redis» - реляционной БД.

Таблица оценки характеристик разных реализаций

|  |  |  |  |
| --- | --- | --- | --- |
| Показатель | dataService | | |
| IPCDataService | DBMDataService | RedisDataService |
| Скорость чтения | линейная | линейная | константная |
| Скорость записи | линейная | квадратичная | константная |
| Загрузка процессора | 25% | 1% | 27% |

По оценке необходимо хранить от 100\_000 до 1\_000\_000 значений телеметрии за все время полета. Такое большое количество телеметрии сильно сказывается на производительности первых двух реализаций, поэтому для работы с телеметрией была выбрана реализация с реляционной БД Redis.

Стоит отдельно упомянуть о том, что данные о загрузке процессора получены на нагрузочных тестах с максимальным количеством операций с сервисами. Загрузка процессора зависит от количества операций с БД. Redis, являясь наиболее быстродействующей БД, способна работать на 5-10% от своей максимальной нагрузки и поэтому реальная загрузка процессора будет ниже в 10-20 раз, что делает реальный показатель загрузки процессора Redis сравнимым с DBM.

# Ключи доступа к хранилищу

Для доступа к телеметрии в БД используются ключи, состоящие из 9 байт. Помимо ключей телеметрии существуют ключи для хранения управляющих команд.

Таблица назначения байтов ключей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Байт 0 | Байт 1 | Байт 2,3,4,5 | Байт 6 | Байт 7,8 |
| Телеметрия подсистемы | Идентификатор  подсистемы | 0x3A | Время добавления в БД (сек) | 0x3A | Время добавления БД (мс) |
| Актуальная телеметрия подсистемы | Идентификатор  подсистемы | 0x3A | 0x6C (‘l’) |  |  |
| Команды подсистемы | Идентификатор  подсистемы | 0x3A | 0x63 (‘c’) |  |  |
| Команды системы | 0x01 | 0x3A |  |  |  |

# Камера

Камера делает одну фотографию каждые две секунды. Для доступа к камере используется библиотека Raspicam. Она возвращает изображение в формате ppm. Далее изображение конвертируется в bmp с помощью адаптированного под данную задачу кода из библиотеки ppm2bmp и сохраняется в папку /home/pi/rcs/img в виде номера изображения и расширения (например «21.bmp»).

Телеметрия камеры включает в себя номер изображения, который сохраняется после сохранения фотографии. Таким образом можно узнать о наличии фотографий по телеметрии камеры.

При перезагрузке системы счетчик номеров фотографий сбрасывается и, для того чтобы его восстановить, модуль камеры при запуске проверяет файлы в папке с фотографиями. Он устанавливает начальное значение счетчика на один больше, чем у фотографии с самым большим номером.

# Критический контроллер

Для общения с КК используется две линии: одна на ввод, другая на вывод. БК чередует каждые 50мс повышающий и понижающий сигналы на своей выходной линии и принимает сигналы по входной линии.

Телеметрия КК состоит из числа импульсов за прошедшую секунду. Импульсы считаются по изменению состояния порта. Сохраняется (кол- во смен состояний)/2.

Ресет малины(юля)

# Приемопередатчик

Данный модуль находится на стадии разработки.

Приложение отправляет на Землю актуальную телеметрию раз в 2 минуты, запрашивая телеметрию из хранилища. Передача данных с помощью ПП имеет ряд особенностей, которые изложены ниже.

Для минимизации расхода электроэнергии простой ПП должен составлять 90- 99% времени работы. Для этого приложение контролирует время передачи пакетов и запускает режим простоя, если лимит времени превышен.

Пакеты телеметрии разного типа имеют разную длину и поэтому, чтобы передавать данные каждого типа отдельно от других, необходимо иметь возможность отправлять пакеты произвольной длины. Такую возможность предоставляет библиотека для работы с ПП.

Для удобства администрирования системы, и на НС, и на БК используются одни и те же ключи базы данных для идентификации пакетов. Таким образом необходимая информация пакета включает в себя ключ телеметрии и саму телеметрию.

# Протокол связи с Землей