

Распределённая система управления антенным комплексом

Елена Петрушкова, Сергей Чадов, Алексей Морозов, Марат Мусин

Представлена распределённая система управления многоканальным радиотехническим комплексом в реальном масштабе времени. Описаны аппаратная и программная архитектура, особенности реализации.

ВВЕДЕНИЕ

Несмотря на высокую степень автоматизации создаваемых радиотехнических комплексов, в эксплуатируемых на сегодняшний день антенных системах продолжают использоваться методы управления, основанные на ручной настройке, наведении на источник излучения и диагностике оборудования, входящего в антенную систему составными частями.

Представленный в данной статье масштабируемый аппаратно-программный комплекс (МАПК) «Союз» обеспечивает управление многоканальным радиотехническим комплексом. Комплекс разработан специалистами ЗАО «КИА Системы».

Под каналом радиотехнического комплекса разработчики понимают антенную систему с автономной системой наведения. Каждый канал управляется единой системой управления и использует общие ресурсы МАПК: систему технического анализа, систему планирования работ, систему долговременно-

го хранения и статистической обработки данных, систему управления контрольно-измерительным оборудованием.

Структура МАПК представлена на рис. 1. Из структурной схемы видно, что МАПК объединяет в одно целое контрольно-измерительную аппаратуру, систему управления исполнительными устройствами, интеллектуальное технологическое оборудование, операторские пункты и другие компьютерные системы, функционирующие в рамках радиотехнического комплекса.

МАПК решает следующие задачи:

- настройка режима работы отдельной антенной системы;
- настройка высокочастотного тракта каждой антенны на оптимальный приём/передачу сигнала в нескольких частотных диапазонах;
- отображение спектра принимаемого сигнала от каждой антенны в любом из диапазонов;
- наведение антенны в заданную точку пространства в автоматическом и ручном режимах управления;

- хранение параметров настройки и наведения по каждой антенне в базе данных АПК;

- контроль параметров антенной системы в целом и каждой входящей в радиотехнический комплекс антенны в реальном масштабе времени.

Разработанный аппаратно-программный комплекс обеспечивает возможность управления целым семейством антенных систем в автоматическом режиме без вмешательства оператора.

ОБОБЩЁННАЯ СТРУКТУРА АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА

Комплекс имеет трёхуровневую иерархическую структуру (см. рис. 1).

Первый уровень представлен системой планирования работ МАПК и системой статистической обработки и долговременного хранения данных.

Второй уровень – это система управления каналами МАПК и системы, реализующие алгоритмы анализа сигналов, контроля и диагностики аппаратуры, а также система управления контрольно-измерительным оборудованием, входящим в состав радиотехнического комплекса. При необходимости расширения круга задач, решаемых радиотехническим комплексом, второй уровень может быть дополнен и другими вычислительными системами.

Третий уровень МАПК образован распределённой системой управления, построенной на базе одноплатных компьютеров CPC303 (компания FASTWEL) формата PC/104, и контрольно-измерительной аппаратурой. Распределённая система управления



Антенны космической связи

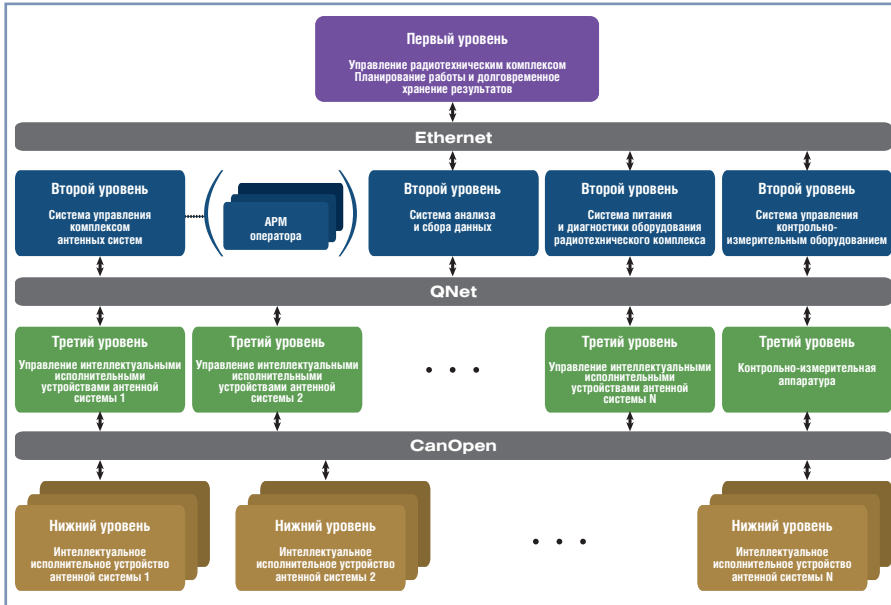


Рис. 1. Структурная схема МАПК

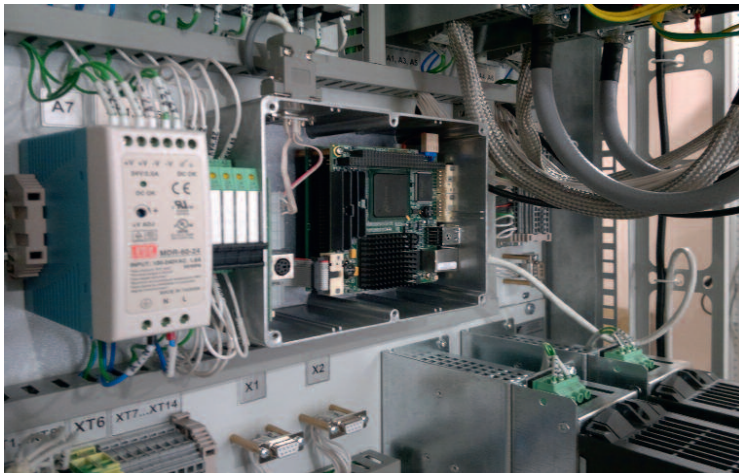


Рис. 2. Управляющий контроллер

имеет классическую для таких систем структуру, объединяющую несколько управляющих контроллеров, в роли которых выступают одноплатные компьютеры CPC303. Внешний вид управляющего контроллера представлен на рис. 2. Контроллеры осуществляют управление интеллектуальными исполнительными устройствами антенных систем в среде встроенной операционной системы реального времени QNX 6 Neutrino. Каждый контроллер управляет работой одного канала, а также осуществляет связь с другими конт-

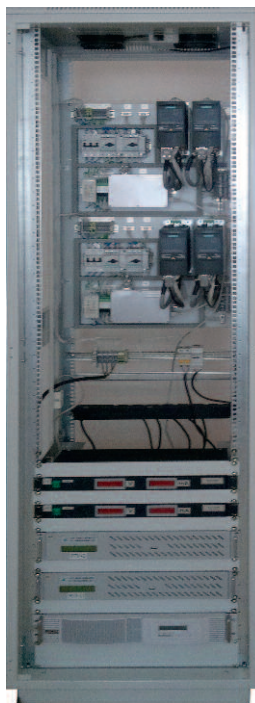


Рис. 3. Аппаратура системы управления для двух каналов комплекса

роллерами и обмен с системами верхнего уровня. Внешний вид исполнения аппаратуры системы управления для двух каналов показан на рис. 3.

Основу нижнего уровня составляют интеллектуальные исполнительные устройства: преобразователи частоты Siemens MICROMASTER 440, энкодеры Kübler, двигатели постоянного и переменного тока, преобразователи и др. В качестве транспортного протокола связи с исполнительными устройствами используется протокол интерфейса CAN, обеспечивающий безопасную передачу данных и команд в реальном масштабе времени.

УПРАВЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫМИ ИСПОЛНИТЕЛЬНЫМИ УСТРОЙСТВАМИ

Функционирование третьего уровня системы основано на межпроцессном взаимодействии посредством передачи сообще-

ний. Процессы могут быть разделены на три группы:

- процессы, реализующие обмен с верхним уровнем системы;
- процессы, реализующие основные задачи управляющих контроллеров;
- процессы, реализующие транспортный протокол с исполнительным интеллектуальным уровнем системы.

Разделение и взаимодействие процессов, которые обеспечивают основную работу по управлению устройствами, обладающими CAN-интерфейсом и имеющими возможность взаимодействовать по нему с другими устройствами, иллюстрирует рис. 4.

Все процессы взаимодействуют через программу управляющего контроллера Coordinator. Обмен со вторым уровнем системы осуществляется по

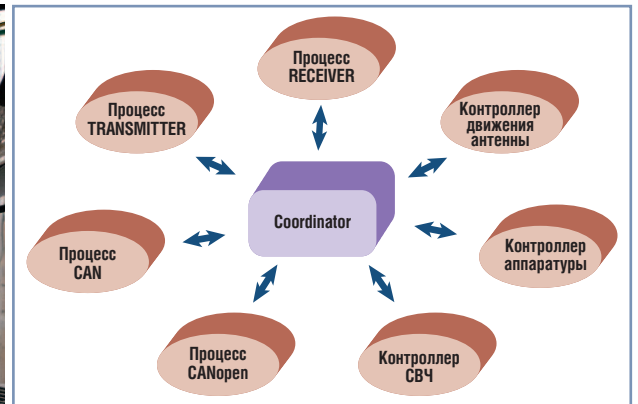


Рис. 4. Взаимодействие процессов

роллерами и обмен с системами верхнего уровня. Внешний вид исполнения аппаратуры системы

протоколу QNet с использованием сокетов и реализуется посредством двух процессов: процесса приёма данных (RECEIVER) и процесса передачи данных (TRANSMITTER). Они обеспечивают непосредственную приёмопередачу данных от систем второго управляющего уровня и передачу их потребителям. Это позволяет организовать асинхронный режим обмена, что не приводит к блокировке систем второго уровня в то время, пока осуществляется передача или приём данных.

QNet позволяет связать компьютеры в однородную сеть, функционирующую на базе собственного одноимённого протокола, основанного на обмене сообщениями микроядра.

Построение сети QNet по сути своей уникально: вместо передачи условных данных, определённых протоколом, по сети передаются системные сообщения, которые после «сборки» менеджером сети поступают непосредственно в ядро. Передача сообщения по сети происходит с использованием механизма, в точности аналогичного тому,

который используется при выполнении запросов системного API, например open(), на локальном компьютере. QNet обеспечивает обмен сообщениями между процессами на удалённых узлах прозрачно, как если бы они находились на одном узле сети, — в результате вся сеть превращается в единый компьютер с общими ресурсами и задачами.

Каждый процесс, реализуемый в контроллере, представляет собой цикл, в котором происходит приём и обработка сообщений от процессов уровнем выше. Построение цикла видно из листинга 1. Все процессы являются потомками абстрактного класса Active, в котором представлены все основные функции: создание канала, по которому идёт приём сообщений, приём сообщений по каналу, обработка сообщений, ответ на сообщения (листинг 2).

Функция MsgReceive является блокирующей. На ней процесс ждёт сообщения, которое предназначено для него. Как только такое сообщение поступает, выполняются анализ сообщения и дальнейшая обработка. Функция MsgReply производит ответ на присы-

лаемые сообщения. Ответ может как содержать сообщение с полезными данными, запрашиваемыми клиентом, так и быть пустым.

Процесс уровнем выше посылает сообщения с помощью функции MsgSend. Но для удобства создан класс IActive (листинг 3). В этом классе реализованы функции для отправки сообщений классу Active. Построение процессов с использованием классов Active и IActive позволяет упростить и ускорить разработку процессов, так как в этих классах уже реализованы основные наборы функций, а разработка каждого процесса (написание соответствующего программного кода) может вестись различными людьми.

Процессы, реализующие основные задачи управляющих контроллеров и обеспечивающие формирование управляющих воздействий для исполнительных устройств нижнего уровня, взаимодействуют между собой через программу Coordinator, которая распределяет ресурсы контроллера между процессами и синхронизирует их работу. Такая орга-

низация позволяет нижнему уровню системы функционировать независимо от второго управляющего уровня и не блокировать задачи, выполняемые системами второго уровня.

СИСТЕМНЫЙ КООРДИНАТОР

На втором уровне системы управления МАПК находятся:

- система управления комплексом антенных систем;
- система анализа и сбора данных;
- система питания и диагностики оборудования;
- система управления контрольно-измерительным оборудованием.

Каждая система, кроме первой из перечисленных, решает свой круг задач и может работать независимо от других. Система управления комплексом антенных систем фактически является системным координатором, менеджером программно-аппаратных средств описываемого радиотехнического комплекса.

ЛИСТИНГ 1

```
while (is_srr_running) {
    rcvid = MsgReceive (chid, r_msg, rsize, &info);
    if (rcvid == -1) {
        warn ("MsgReceive error");
        continue;
    }

    if (rcvid != 0) {
        ProcessMsg (rcvid, r_msg, &info, &status, p_msg, &psize);
        if (status != -1) {
            rc = MsgReply (rcvid, status, p_msg, psize);
            if (rc == -1)
                warn ("MsgReply error");
        }
    } else
        ProcessPulse (r_msg, &info);
}
```

ЛИСТИНГ 2

```
class Active {
public:
    Active ();
    virtual ~Active ();
    void Run (int argc, char** argv);

protected:
    virtual void ProcessMsg (int rcvid, void* r_msg,
        struct _msg_info* info, int* status, void* p_msg, int* psize);
    virtual void ProcessPulse (void* r_msg, struct _msg_info* info);

    void MsgEmpty (int rcvid, void* r_msg, struct _msg_info* info,
        int* status, void* p_msg, int* psize);
    void PulseEmpty (void* r_msg, struct _msg_info* info);

private:
    void MainLoop (void* r_msg, void* p_msg, int chid, int rsize);
    void MsgQuit (int rcvid, void* r_msg, struct _msg_info* info,
        int* status, void* p_msg, int* psize);
    void PulseDisconnect (void* r_msg, struct _msg_info* info);
    void PulseQuit (void* r_msg, struct _msg_info* info);
};
```

ЛИСТИНГ 3

```
class IActive {
public:
    IActive (int ssize, int lsize);
    ~IActive ();

    int Open (const char* name, int flags);
    int Close ();
    int Completing ();
    int PulseCompleting ();
};
```

ЛИСТИНГ 4

```
class Device
{
public:
    Device();
    Device(<список параметров>);

    //Выбор частоты опроса устройства
    unsigned int requestInterval() const;
    void setrequestInterval (unsigned int interval);

    //Установка и запрос сетевого адреса устройства
    string hostAddress() const;
    void setHostAddress (const string &address);

    //Установка и запрос порта устройства
    unsigned int port();
    void setPort (unsigned int port);

    /*Здесь описываются методы для установки специфичных
    для устройств параметров
    */

private:
    string          hostAddress;      //Сетевой адрес устройства
    unsigned int    port;             //Порт устройства
    int             socket;           //Дескриптор сокета

    /*Описание остальных параметров устройства
    */

    void initialize();                //Инициализировать устройство
    void sendCommand(const char *)    //Отправить команду устройству
    char *response();                 //Прочитать ответ от устройства
    void getData();                   //Считать данные (значение)
    void start();                      //Запустить цикл опроса
    void stop();                       //Завершить цикл опроса
};
```



Рис. 5. Главный экран системы управления комплексом

Основной задачей этой системы является планирование ресурсов МАПК и обеспечение взаимодействия систем различного назначения, входящих в него, а также реализация наглядного и детального представления всех частей МАПК, обеспечение ясной обратной связи и соблюдение эргономических требований к графическому интерфейсу. В процессе решения данных задач удалось спроектировать интерфейс, который позволяет максимально точно управлять комплексом, своевременно сообщает оператору обо всех неисправностях и учитывает специфику предметной области для создания комфортных условий использования. Копия основного экрана системы управления комплексом приведена на рис. 5.

Поскольку система управления МАПК позиционируется как многоканальная унифицируемая система с возможностями расширения, целесообразно было предусмотреть механизм, позволяющий наращивать её независимо от уже существующей и функционирующей структуры. Поэтому был разработан механизм, который обеспечивает возможность быстрой доработки системы при условии появления дополнительного оборудования, включаемого в контур управления. С этой целью разработан объект типа «класс», позволяющий описывать поведение объекта управления. В листинге 4 приведён каркас программного кода класса Device. Представленный класс предназначен для работы с оборудованием по протоколу TCP/IP, поэтому для установления соединения необходимо задать адрес и порт устройства. Метод sendCommand() выполняет работу по формированию сообщения, проверке

его корректности и отправке сообщения устройству – иными словами, обеспечивает требуемый протокол. Этот метод используется как основа для более специфичных методов класса, например таких как getData(), который предназначен для считывания контролируемых параметров с устройства.

Вся работа с устройством осуществляется в цикле опроса, который описывается индивидуально для разных устройств. Внутри него мы считываем данные и осуществляем их обработку. Стандартный цикл предполагает опрос с выбранным временным интервалом. Если же требуется настроить цикл опроса более детально, то можно переопределить метод start().

Для добавления нового оборудования в контур управления комплексом достаточно внести в представленный

класс лишь специфичные для данного типа оборудования изменения. Такой подход позволяет значительно сократить время на интеграцию нового оборудования и делает доработку программного обеспечения независимым процессом, допуская привлечение программистов разной квалификации, что в конечном итоге может значительно удешевить процесс разработки программного обеспечения.

ЗАКЛЮЧЕНИЕ

Описанный в данной статье аппаратно-программный комплекс со своей системой управления введён в эксплуатацию в составе многолучевого антенного комплекса и успешно функционирует.

Поскольку разработчики данной системы ставили своей основной целью получить динамически развивающийся инструмент, позволяющий быстро адаптировать программное обеспечение к различным объектам, на его основе может быть выстроена система управления реального времени, позволяющая управлять различными объектами промышленного и специального назначения.

В качестве примеров возможного применения описанной в статье системы управления можно привести различные гексаподы, используемые в гиросtabilизированных платформах (электронный гиросtabilизатор), робототехнике (управление сложными движениями), шлюзах и других приложениях. ●

E-mail: e-pet@yandex.ru

НОВОСТИ НОВОСТИ НОВОСТИ НОВОСТИ

Компания Pepperl+Fuchs получила Reader's Choice Award в четырёх категориях продукции

Журнал Control Design учредил награду Reader's Choice Award («Выбор читателей») в 2001 году. Компания Pepperl+Fuchs получила этот замечательный знак отличия десять лет подряд в разных товарных категориях. В 2011 году компания Pepperl+Fuchs была выбрана заказчиками за отличное качество продукции в четырёх категориях.



Компания получила оценку заказчиков 4,3 балла (при максимальной оценке 5) в категориях искробезопасных систем и оборудования с видом взрывозащиты «заполнение или продувка оболочки под избыточным давлением защитным газом» и 4,1 балла в категории компонентов для сетевой безопасности. Клиентские оценки работы Pepperl+Fuchs располагаются между «очень хорошо» (4) и «отлично» (5).

Категории, в которых Pepperl+Fuchs получила награды, приведены в таблице. ●

Место в категории	Название категории	Количество голосов, %
1	Промышленные системы с заполнением или продувкой оболочки под избыточным давлением	39
1	Искробезопасные компоненты	20
7	Датчики присутствия	6
3	Компоненты для сетевой безопасности	11