

# КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

V. Zinchenko, V. Tixomirov,  
V. Zinchenko, A. Potapenko,  
N. Brovarska

## **THE PROBLEM OF MODELING HARDWARE AND SOFTWARE ENVIRONMENT OF THE COMPLEX OF AVIONICS FOR STAGES OF DESIGN, TESTING AND OPERATION**

*It was considered urgent tasks and mediation in the implementation of the control of currents in multiplexed traffic channel for a multi-level hardware and software environment onboard avionics.*

*Key words: object, data, information flows, the operator.*

*Розглянуті особливості, актуальні задачі і засоби реалізації керування струмами в мультиплексному каналі інформаційного обміну для багаторівневого апаратно-програмного середовища бортового комплексу авіоники.*

*Ключові слова: об'єкт, інформаційні потоки, оператор.*

*Рассмотрены особенности, актуальные задачи и средства реализации управления потоками в мультиплексном канале информационного обмена, для многоуровневой апаратно-программной среды бортового комплекса авионики.*

*Ключевые слова: объект, информационные потоки, оператор.*

© В.П. Зинченко, В.А. Тихомиров,  
С.В. Зинченко, А.Ф. Потапенко,  
Н.И. Броварська, 2016

УДК 629.735.05:681.326.3

В.П. ЗИНЧЕНКО, В.А. ТИХОМИРОВ, С.В. ЗИНЧЕНКО,  
А.Ф. ПОТАПЕНКО, Н.И. БРОВАРСКАЯ

## **ЗАДАЧИ МОДЕЛИРОВАНИЯ АППАРАТНО-ПРОГРАММНЫХ СРЕД КОМПЛЕКСОВ АВИОНИКИ ДЛЯ ЭТАПОВ ПРОЕКТИРОВАНИЯ, ИСПЫТАНИЙ И ЭКСПЛУАТАЦИИ**

**Введение.** Переход от аналоговых к цифровым средствам авионики привел к тому, что реализация функций переместилась из области схмотехники в область программирования. Это касается и мультиплексного канала информационного обмена (МКИО) систем с центральным управлением или интерфейса Mil-Std-1553B (стандарт США) [1]. Используется в авионике и во всех видах вооружений США и стран НАТО. Аналоги: ГОСТ 26765.52-87 – СССР; ГОСТ P52070-2003 – Россия; DefStan 00-18/2 – Британия; STANAG 3838 – НАТО; AirStandart 50/2 – комитет ASCC [2].

Широкое применение МКИО Mil-Std-1553B объясняется следующим: линейная топология; малая масса и габариты, гибкость в подключении/отключении устройств; информационная надежность (автоматическая передача информации при отказе по резервной шине); детерминизм (работа в реальном времени (РВ)); поддержка неинтеллектуальных терминалов (датчики, исполнительные устройства); устойчивость к отказам (стабильная работа при отказе любого устройства благодаря автоматической реконфигурации); доступность компонентов.

Предложенная моделирующая система обеспечивает организацию режима виртуального окружения для этапов проектирования, испытаний и эксплуатации, а также коплексирования апаратно-программных сред компонентов авионики.

**Основная часть.** В работе модель бортового комплекса авионики (МБКА) рассматривается как многомашинная многоуровневая система (на рис. 1 показан один уровень), где  $SS_i$  /SubSystem – подсистемы (индикации, управления, навигации, противообледенения и т. п.), которые взаимодействуют между собой посредством интерфейса МКИО через устройства интерфейса (УИ), которые работают или как удаленный терминал – Remote Terminal (RT), или как контроллер шины – Bus Controller (BC) или как монитор шины – Bus Monitor (BM). Отметим, что реально вместо персонального компьютера (ПК) может быть штатная бортовая система технического обслуживания (БСТО) и/или система бортовых измерений (СБИ) для этапа испытаний, т. е. одна из  $SS_i$ , например,  $SS_m$ . У одной подсистемы может быть несколько устройств интерфейса и наоборот: несколько подсистем могут быть подключены к одному УИ.

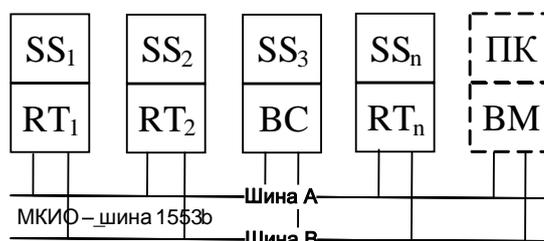


РИС. 1. Схема типового уровня МБКА

Основные задачи интерфейса Mil-Std-1553В две: организация обменом информации между подсистемами авионики (через устройства интерфейса шины); управление обменом (контроль передачи/приема информации, состояния УИ шины; обработка отказных ситуаций).

Прикладные функции интерфейса реализуются прикладными программами (ПП), так как схемотехническая реализация стандартизована в виде электронных компонентов. Существуют устройства, позволяющие моделировать информационные ситуации. Например, устройства CM1553-1/3 [3, 4] фирмы Ballard Technology (США) позволяют не только освоить функции интерфейса, но и создавать полунатурные модели процессов взаимодействия подсистем авионики на этапах проектирования, испытаний, отработки и эксплуатации.

Главным активным организующим и управляющим элементом в любой подсистеме является тот терминал шины, который в данный момент времени работает в режиме BC. Терминалы в режиме RT выполняют адресованные им команды BC. Отметим, что количество МКИО можно увеличить, объединив их в иерархическую структуру. Терминал в режиме BM не выполняет команд BC, а непрерывно контролирует появляющуюся в шине данных информацию. Режимы работы задаются соответствующими функциями драйвера устройства CM1553-1.

В работе исследованы типовые базовые задачи для МБКА.

**Задача 1.** Передать одно сообщение один раз от ВС к RT или наоборот.

Алгоритм решения задачи передачи в прямом направлении (от ВС к RT) в LAC-форме [3] записи сообщений в МКИО будет таким:

$$CW:AD.TR.SA.WC+DW[1]:WV[1][+DW2:WV2]...[+DWn:WVn]= \\ =T1+SW:AD.SB[.SB]+T2;$$

или в сокращенном виде

$$CW:AD.TR.SA.WC+DW:\{n\}=T1+SW:AD.SB[.SB]+T2;$$

где CW/Command Word – командное слово; поля CW (AD/address – адрес RT, которому предназначено сообщение; TR/Transmit-Receive bit – значение бита передача/прием; SA /SubAddress – подадрес параметра передаваемых данных; WC/Word Count – количество передаваемых DW); DW/Data Word – слово данных; WV/Word Value – значение DW; n – количество реально переданных DW (без ошибок n = WC, с ошибками n ≠ WC); SW/Status Word – слово состояния (ответное слово) RT; SB/Status Bit – номер бита признака SW при значении = 1; T1 и T2 – паузы,  $4\text{мкс} \leq t_1 \leq 12\text{мкс}$  и  $t_2 \geq 4\text{мкс}$  в передаче.

Таким образом, ВС должен передать одно CW и необходимое число DW после чего ожидать заданное время ответа от RT в виде SW.

Алгоритм передачи информации в обратном направлении (от RT к ВС) следующий:

$$CW:AD.TR.SA.WC=T1+SW:AD.SB[.SB]+DW[1]:WV[1][+DW2:WV2].....[+DWn:WVn]+T2;$$

или в сокращенном виде

$$CW:AD.TR.SA.WC=T1+SW:AD.SB[.SB]+DW:\{n\}+T2.$$

Следовательно, ВС должен передать одно CW, после чего ждать заданное время ответа от RT в виде SW и необходимого числа DW. Соответствующий фрагмент ПП для ВС имеет вид:

```
...
#include<C11C.h>
handle=C11_OpenCard (socketval);
command=C11_PackCWD (ad, tr, sa, wc);
<определить и установить значения передаваемых данных в data,
если направление передачи ВС-RT>
C11_TransmitMsg (BUSA, command, data, RTstatus, handle);
<обработать значение RTstatus полученного слова состояния RT
и определить дальнейший путь ПП, обработать значения
полученных данных, если направление передачи RT-ВС>
C11_CloseCard(handle);
...
```

где C11C.h – библиотека функций; handle – дескриптор устройства CM1553-1; C11\_OpenCard (socketval) – функция подключения CM1553-1 в гнезде PCMCIA (socketval – номер гнезда); command – возвращаемое CW; C11\_PackCWD (ad, tr, sa, wc) – утилита создания CW из полей; C11\_TransmitMsg (BUSA, command, data, RTstatus, handle) – функция установки CM1553-1 в режим ВС и передачи

данных между ВС и RT в зависимости от бита TR; BUSA /B – определить обмен по шине А или В; C11\_CloseCard (handle) – функция отключения CM1553-1.

**Задача 2.** Периодическая передача сообщений/группы данных (фрейм) ВС по шине. Например, необходимо передавать такие два сообщения с частотой 50 Гц:

сообщение 1: RT с AD = 2 должен принять от ВС WC = 8 DW в SA = 1;

сообщение 2: RT с AD = 4 должен передать к ВС WC = 12 DW из SA = 1.

Один из способов решения задачи – это размещение функции C11\_TransmitMsg(\*) в цикл с задержкой по времени. Этот способ имеет такие недостатки: бесосновательная загрузка ПК; возможные проблемы синхронизации. Лучший способ – это автоматическая синхронная передача фрейма с использованием отдельных функции. Алгоритм (сокращенный вид) выглядит следующим образом:

CW:02.0.01.08 = T1 + SW:02.00 + DW:{08} + T2;

CW:04.1.01.12 = T1 + SW:04.00 + DW:{12} + T2.

Соответствующий фрагмент ПП для ВС имеет такой вид:

```
...
Handle = C11_OpenCard (0);
C11_BCConfig (CONFIG_DEFAULT, handle);
C11_BCFrameStart (20, handle); /*20 – период передачи в мс*/
msg1 = C11_BCDoMsg (BUSA, 1028h, dataout, handle);
msg2 = C11_BCDoMsg (BUSA, 242Ch, datain, handle);
C11_BCFrameEnd(handle); /*1028h и 242Ch – значения двух CW*/
C11_Start(handle);
C11_BCRdData (msg2, datain, msg2count, handle);
<обработка входных данных datain>
<изменение выходных данных dataout, когда требуется в ПП>
C11_BCWrData (msg1, dataout, msg1count, handle);
C11_Stop(handle);
C11_CloseCard(handle);
...
```

Открывается CM1553-1 функцией C11\_OpenCard, конфигурация устройства устанавливается функцией C11\_BCConfig(\*) и описывается фрейм – между функциями C11\_BCFrameStart(\*) и C11\_BCFrameEnd(\*). После активизации функцией C11\_Start(\*) устройство CM1552-1 выполняет передачу фрейма в режиме повторения. При этом непосредственно перед передачей первого сообщения фрейма задается время его жизни (в примере, 20 мс). CM1553-1 передает сообщения в нужном порядке, затем ожидает заданное время и повторяет передачу фрейма.

Функции C11\_BCWrData(\*) и C11\_BCRdData(\*) используются для инициализации и управления данными (массивы datain и dataout). Поскольку CM1553-1 работает независимо от ПК, то одни и те же данные повторяются в каждом со-

общении до их изменения функцией C11\_BCWrData(\*). Последние принимаемые данные сохраняются и их можно прочитать функцией C11\_BCRdData(\*).

**Задача 3.** Периодическая передача ВС списка сообщений (циклограмма). Например, необходимо передать три сообщения (msg1 с частотой 100 Гц, msg2 с частотой 50 Гц, msg3 с частотой 25 Гц) и на неисправность RT (передача msg1 закончилась аварийно) должен отреагировать ВС путем передачи специального сообщения (errormsg).

Соответствующий фрагмент ПП имеет такой вид:

```

...
C11_BCConfig (CONFIG_DEFAULT, handle);
/* аperiodические сообщения */
errormsg=C11_BCDoMsg (BUSA, errorcommand, errordata, handle);
C11_BCBranchEnd(handle);
/* фреймы */
frame1 = C11_BCFrameStart (10, handle);
msg1 = C11_BCDoMsg (BUSA, command1, data1, handle);
C11_BCMsgBranch (msg1, errormsg, COND_RESPERR, handle);
msg2 = C11_BCDoMsg (BUSA, command2, data2, handle);
msg3 = C11_BCDoMsg (BUSA, command3, data3, handle);
C11_BCFrameEnd (handle);
frame2 = C11_BCFrameStart (10, handle);
C11_BCDoAgain (msg1, handle);
C11_BCFrameEnd(handle);
frame3 = C11_BCFrameStart (10, handle);
C11_BCDoAgain (msg1, handle);
C11_BCDoAgain (msg2, handle);
C11_BCFrameEnd(handle);
/* цикл */
C11_BCCallFrame (frame1, handle);
C11_BCCallFrame (frame2, handle);
C11_BCCallFrame (frame3, handle);
C11_BCCallFrame (frame2, handle);
C11_Start(handle).

```

Отметим, что аperiodические сообщения могут обслуживать не только несколько случаев отдельного сообщения (как в примере), но и несколько различных сообщений.

**Задача 4.** Передача сообщений между RT. Например, необходимо передавать периодическое сообщение с частотой  $f = 1/\text{time}$  от RT1 с AD1 к RT2 с AD2, т. е. передать несколько (WC) DW из SA1 в SA2. Алгоритм:

$$\begin{aligned}
 & CW1:AD2.0.SA2.WC + CW2:AD1.1.SA1.WC = \\
 & = T1 + SW1:AD1.SB[.SB] + DW:\{n\} + T1 + SW2:AD2.SB[.SB] + T2,
 \end{aligned}$$

где  $n = WC$ .

Другими словами, ВС должен передать 2 CW: первое в адрес RT2 на прием информации, второе в адрес RT1 на ее передачу, после чего ожидать от них отве-

тов в виде SW от RT1 с необходимым количеством DW, затем SW от RT2. Соответствующий фрагмент ПП для BC такой:

```
...
Handle = C11_OpenCard (socketval);
  C11_BCConfig (CONFIG_DEFAULT, handle);
    cmd1 = C11_PackCWD (AD2, 0, SA2, WC);
    cmd2 = C11_PackCWD (AD1, 1, SA1, WC);
  C11_BCFrameStart (time·10-3, handle);
  C11_BCDoRTRT (BUSA, cmd1, cmd2, handle);
  C11_BCFrameEnd (handle);
  C11_Start (handle);
...

```

**Задача 5.** Режим RT. Так как RT только выполняет команды BC шины, то список сообщений не входит в элементы его конфигурации.

Пусть RT с AD = 3 связаны два сообщения – одно на прием, второе на передачу. Используя CW = 1828h BC предлагает RT принять 8 DW в SA = 1, а с помощью CW = 1C24h – передать 4 DW из SA1. В этом случае ПП должна обеспечить считывание и обработку принятых DW, модификацию и запись данных для передачи. Алгоритм сообщений будет таким:

CW:03.0.01.08+DW:{08}=T1+SW:03.00+T2;

CW:03.1.01.04=T1+SW:03.00+DW:{04}+T2.

Соответствующий фрагмент ПП для BC:

```
...
handle = C11_OpenCard (0);
  C11_RTConfig (CONFIG_DEFAULT, address, handle);
  C11_RTWrData (1C24, dataout, handle);
  C11_Start(handle);
  C11_RTRdData (1828h, datain, handle);
<обработка полученных входных данных datain>
<изменение выходных данных dataout для передач>
  C11_RTWrData (1C24h, dataout, handle);
  C11_Stop (handle);
  C11_CloseCard (handle);
...

```

Важным параметром конфигурации является адрес RT, назначаемый функцией C11\_RTConfig(\*), так как CM1553-1 отвечает только на CW, содержащих указанный адрес. После инициализации области передачи данных (функция C11\_RTWrData(\*)) и перевода устройства в активное состояние (функция C11\_Start(\*)), CM1553-1 может передавать/принимать данные в ответ на CW с адресом RT (в нашем случае address=3).

Функции C11\_RTRdData(\*) и C11\_RTWrData(\*) управляют данными (массивы datain и dataout). Когда CM1553-1 принимает достоверные команды и DW, то данные сохраняются во внутренней памяти по назначенным для приема SA.

При необходимости с помощью функции `C11_RTRdData(*)` можно получить эти данные по заданному SA. Когда устройство CM1553-1 получает достоверную команду на передачу, то оно выдает ответное SW и указанное число DW из конкретного SA. Для каждой команды на передачу с заданным SA передаются те же данные до тех пор, пока они не будут изменены функцией `C11_RTWrData(*)`.

**Задача 6.** Режим VM: устройство CM1553-1 записывает сообщения и размечает время всего обмена по шине для заданных адресов RT.

Фрагмент ПП записи 512 сообщений будет таким:

```
...
index = 0;
handle = C11_OpenCard (0)
  C11_MONConfig (CONFIG_DEFAULT, 512, handle);
  C11_Start(handle);
  while (index < 512);
  if (index = C11_MONCount(handle));
  C11_MONRd (index, block, data, handle);
<обработка массивов block и data, приращение index>
  C11_Stop(handle);
  C11_CloseCard(handle);
...
```

Функция `C11_MONConfig(*)` определяет конфигурацию на запоминание 512 сообщений и производит мониторинг всех RT шины. Для мониторинга конкретного RT используется функция `C11_MONTermAddr(*)`.

Функция `C11_MONCount(*)` возвращает номер последнего сообщения, принятого устройством CM1553-1, как значение переменной `index`. Если `index` меньше чем значение, возвращенное `C11_MONCount(*)`, то `C11_MONRd(*)` читает следующее сообщение с CM1553-1. Отметим, что временные метки, информация об ошибках и CW, SW сохраняются в массиве `block`; а DW – в массиве `data`. ПП могут обрабатывать `block` и `data` согласно алгоритму прикладной задачи, например, сохранить информацию на диске для дальнейшего использования.

После того, как необходимые 512 сообщений прочитаны, CM1553-1 переводится в пассивное состояние – функция `C11_Stop(*)`. Если устройство не переведено в пассивное состояние, то оно будет циклически продолжать запись в буфер 512 сообщений, что используется для продолжительной регистрации данных.

Приведенные задачи иллюстрируют только некоторые характерные для Mil-Std-1553B случаи применения устройства CM1553-1. В библиотеке `C11C.h` содержатся 64 функции, которые позволяют решать такие характерные задачи с применения МКIU в авионике:

- ВС: повторная передача сообщений; задание интервалов времени между сообщениями; вызов прерываний на ПК в заданных точках; выполнение встроенного теста;

- RT: буферизация (общий буфер, типа пинг-понг, индексный список); задание признаков занятости, ошибок и неисправностей; блокирование конкретного CW (запрет приема/ передачи); чтение данных с метками времени; вызов прерываний по приему отдельных CW или при ошибках;

- VM: совмещенный режим RT и VM.

Следует отметить, что, имея большое количество устройств СМ1553-1 и ПК, можно построить достаточно сложную адекватную реальной системе авионики моделирующую систему. В тоже время, обычно разработчик, испытатель или исследователь в конкретный момент времени работает с одним/несколькими параметрами информационного обмена, реализующих какую-либо одну функцию. Поэтому для полного моделирования достаточно три-четыре устройства СМ1553-1, что позволит решать все задачи применения МКИО на этапах проектирования, испытаний или эксплуатации авионики. Моделирующая система авионики включает 3 ПК с устройствами СМ1553-1 и шиной 1553В.

На этапе проектирования авионики с применением МКИО наиболее важным является возможность моделирования информационного взаимодействия ее подсистем в РВ. Отметим, что ранее для аналоговых и ранних цифровых систем это в принципе было невозможно, поэтому использовались расчетные методы. Реальная отработка взаимодействия (“стыковка”) производилась на позднем этапе испытаний с использованием построенных экспериментальных образцов. Материальные, временные и финансовые затраты, а особенно с учетом устранения ошибок, были очень велики. Возможность динамического моделирования позволяет отработать не только взаимодействие между подсистемами, но и алгоритмы их функционирования, включая критические ситуации [5].

Особое значение такая моделирующая система имеет на этапе испытаний авионики за пределами эксплуатационных ограничений самолета (безопасность), что невозможно проверить в реальных условиях. Моделирующая система позволяет задать значения/градиенты параметров подсистем за пределами ограничений, а значит позволяет не просто прогнозировать поведение подсистем авионики, но получить реакцию на заданные возмущения, и экспериментально подтвердить реальные границы.

Сертификация критических подсистем требует подтверждение безопасности их эксплуатации при возникновении отказов/неисправностей. Известно, что далеко не все отказы/неисправности, можно имитировать на реальных системах, например, деградацию параметров. И здесь единственным решением задачи является моделирование.

Моделирующая система является инструментом в условиях жесткого РВ обнаружения и устранения неисправностей подсистем авионики в условиях эксплуатации. Контрольно-поверочная аппаратура, применяемая до сих пор, не дает желаемого результата, поскольку реализует фиксированные точки или статичные ситуации в области функционирования. Универсальный инструмент, каковым является ПК с соответствующим ПО, позволяет моделировать практически любые ситуации в РВ.

Одним из полезных с практической точки зрения применений стандарта Mil-Std-1553В является запись на внешние носители в режиме ВМ всех сообщений по МКИО в течение всего полета самолета. По окончании полета наземные испытательные/эксплуатационные службы используя режимы ВС и RT могут неограниченное количество раз воспроизвести этот полет в РВ или непосредственно на бортовых экранах на рабочих местах членов экипажа, или на экране ПК с моделью системы индикации. Другими словами, воспроизвести приборную обстановку работы экипажа, с точностью до бита, т. е. до 1 мкс. Можно также создать архив полетов самолета за весь его жизненный цикл, и эти записи использовать для оценки действий экипажа, в том числе, о их влиянии на ресурс самолета.

Известно, что на всех этапах жизненного цикла самолета документация является важным элементом. Следовательно, необходим такой формат представления сообщений в МКИО, который позволял бы, с одной стороны, записывать проектируемые алгоритмы в виде ПП подсистем авионики в терминах понятных программисту и близким к операторам языка программирования, с другой – в терминах, читаемых и понимаемых без дополнительных средств разработчиком/испытателем таких подсистем. Как раз, используемые в примерах алгоритмы задач ЛАС-формы записи отвечают этим требованиям. Например, приведем фрагмент записи сообщений из МКИО реального самолета с некоторыми изменениями, что обусловлено практическим опытом. Во-первых, пауза T1 явно не показывается, если она соответствует стандарту, иначе – выдается сообщение об ошибке. Во-вторых, пауза T2 обозначается в таких случаях: когда необходимо знать резервы времени на асинхронные сообщения, что блокируют неисправности; когда выставляются метки текущего таймера в микросекундах в начале каждого сообщения. Разница между двумя соседними значениями таймера – это общее время передачи одного сообщения, иначе – сумма длин всех слов и всех пауз (одной /двух T1 и одной T2) в сообщении:

$$T_c = T_{CW1} + [T_{CW2} + T_{SW1} + T_{SW2}] [+n \cdot T_{DW}] [+T1 + T1] + T2.$$

В таком случае фрагмент выглядит так:

```
0324096 CW1:31.0.19.18+CW2:12.1.03.18=SW:12.00+DW:{18};
0324608 CW1:31.0.12.12+CW2:15.1.25.12=SW:15.00+DW:{12};
0324992 CW1:31.0.20.02+CW2:11.1.02.02=SW:11.00+DW:{02};
0325184 CW:31.0.06.06;
0325440 W1:13.0.01.02+CW2:07.1.02.02=SW1:07.00+DW:{02}+SW2:13.00;
0325696 CW1:31.0.17.02+CW2:16.1.03.02=SW:16.00+DW:{02};
...
0344128 CW1: (начало следующего цикла).
```

Значения DW извлекаются в цифровом виде. Однако характеристики работы систем и параметры полета самолета имеют физический смысл, и их желательно, а часто необходимо, отображать в виде графиков, диаграмм и т. п.

Моделирующую систему можно также использовать для обучения комплексированию и программированию интегральных комплексов авионики на базе интерфейса 1553В. В этом случае желательно ввести в систему двухлуче-

вой осциллограф для наблюдения физических процессов в шинах МКИО. Одновременно могут работать три оператора как автономно, так и во взаимодействии в РВ [3, 4, 11].

**Выводы.** В результате проведенных исследований типовых базовых задач для МБКА позволили сформулировать подход к созданию унифицированной аппаратно-программной среды, которая обеспечивает решение вопросов комплексирования многоуровневых систем с учетом специфики функциональных подсистем комплекса авионики на этапах проектирования, испытания и эксплуатации. Кроме того, моделирующую систему можно также использовать для обучения комплексированию и программированию интегральных комплексов авионики на базе интерфейса 1553В в режиме виртуального окружения. В этом случае желательно ввести в систему средства визуализации (например, двухлучевой осциллограф) для наблюдения физических процессов в шинах МКИО. Одновременно могут работать три оператора как автономно, так и во взаимодействии в РВ.

1. MIL-STD-1553B. Military Standard. Aircraft Internal Time Division Command. *Response Multiplex Data Bus*. Notice 3. Washington: DC 20360 USA. DODSTD, 1993. 50 p.
2. [What is MIL-STD-1553](http://www.milstd1553.com) // <http://www.milstd1553.com>.
3. CM1553-1 User's Manual. Mil-Std-1553 PCMCIA Card and Software. *Everett, WA98201 USA, Ballard Technology, Inc.* 1995. 76 p.
4. Astronics Ballard Technology - Avionics Databus Solutions. <http://www.ballardtech.com/products.aspx/CM1553-1>.
5. RTCA/DO-178B. Software considerations in airborne systems and equipment certification. Washington: RTCA, 1992. 105 p.

Получено 08.07.2016