

УДК 629.735.05:681.326.3

DOI: <http://dx.doi.org/10.20535/2219-380413201566692>

**Зинченко В. П.**<sup>1</sup>, *доцент, к.т.н.*, **Тихомиров В. А.**<sup>2</sup>, *начальник отдела, к.т.н.*, **Зинченко С. В.**<sup>3</sup>, *научный сотрудник*

## **О ПОДГОТОВКЕ СПЕЦИАЛИСТОВ БОРТОВЫХ КОМПЛЕКСОВ АВИОНИКИ**

**En** The transition from analogue to digital means of avionics led to the fact that the implementation of the functions has passed from the circuitry in the area of programming. This also applies to the multiplex channel of information exchange (MCIE) or interface MilStd-1553b.

The features of the interface and relevant software task, information management in MCIE.

The model of on-board avionics is presented as the multi-machine multi-level system consisting of subsystems (display, control, navigation, de-icing, etc.) which interact with each other through MCIE via the device interface, which can operate as remote terminal, bus controller and monitor of the tire.

Applied interface functions are implemented by applied programs and device modeling situations, for example, CM1553-1/3. This allows us to develop interface functions and create scaled-down model of the processes of interaction between the avionics subsystems in real time (RT). Software is a driver and library functions C11C.h for C/C++ and Borland compilers, Turbo, Microsoft Visual. To illustrate the application of the device CM1553-1 discussed solutions to common problems in LAC-form writing messages to MCIE.

The proposed modeling system for avionics: design (modeling of the information interaction of the subsystems in RT and debugging of the algorithms of their functioning, including critical situations); tests (outside the operational limits of the aircraft that cannot be tested in real conditions); detects and eliminates faults in operational conditions. For example, writing messages on MCIE during aircraft entire flight allows ground-based services to play an instrument the situation of the crew on side screens or on the computer screen.

It was proposed to use the format of the messages in MCIE that allows you to write the algorithms of the avionics subsystems in terms understandable to the programmer and the developer /tester of the subsystems.

Modeling system is proposed to use for learning integration and programming of integrated avionics.

Summary: the use of digital technology in avionics necessitates the training of professionals; experience in avionics MCIE allows you to organize training in the principles of construction, interaction and implementation of avionics components with the use of program-controlled links; the introduction of digital technology in avionics requires the broader and more substantial attention to the problem of learning.

---

<sup>1</sup> *Національний технічний університет України "Київський політехнічний інститут", кафедра автоматизації експериментальних досліджень*

<sup>2</sup> *ГП «Антонов»*

<sup>3</sup> *Інститут кібернетики ім. В. М. Глушкова НАН України*

**Ua**

Розглянуто особливості та актуальні завдання управління потоками інформації у мультиплексному каналі інформаційного обміну (МКИО).

Модель бортового комплексу авіоніки представлена як багатомашинна багаторівнева система, яка включає підсистеми (індикації, управління, навігації, антикригова, ін.), що взаємодіють між собою через пристрої інтерфейсу. Прикладні функції МКИО реалізуються прикладними програмами та пристроями СМ1553-1/3. Наведено приклади застосування при запису алгоритмів у LAC-формі.

Запропоновано моделюючу систему авіоніки для етапів проектування, випробувань і експлуатації, і формат повідомлень для запису алгоритмів в термінах зрозумілих як програмісту, так і розробнику підсистем.

Моделюючу систему запропоновано використовувати для навчання комплексуванню і програмуванню інтегральних комплексів авіоніки.

### **Введение**

Переход от аналоговых к цифровым средствам авионики привел к тому, что реализация функций переместилась из области схемотехники 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 Р52070-2003 – 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.

Внедрять *Mil-Std-1553B* в авионику отечественных самолетов пришлось в условиях отсутствия учебных пособий и лабораторий. Только после наработки определенного опыта использования МКИО в авионике стало возможным говорить об организации подготовки специалистов [3, 4].

### **Актуальность задачи**

Цифровые компоненты авионики обрабатывают большие объемы данных и управляют информационными потоками. Для решения актуальной задачи программного управления потоками информации необходимы средства информационного обмена с такими свойствами: высокая надежность и достоверность передачи информации; высокая

пропускная способность канала; возможность оперативной смены задач (перепрограммирование); унификация компонент; малые массо-габаритные характеристики. Названные требования реализует стандарт *Mil-Std-1553B*, в отличие, например, от документа *ARINC-429* [5].

### Особенности интерфейса

Шина МКИО состоит из бифилярной пары проводов с волновым сопротивлением 70-85 ом с частотой импульсов 1 МГц. Для соединения применяется цилиндрический разъём, по центральному контакту которого передаётся, закодированный Манчестерским кодом сигнал. Оконечные устройства (передатчики /приемники) подключаются к МКИО посредством трансформаторной развязки (каплер); по концам шины устанавливаются согласующие устройства (терминаторы). Все это уменьшает влияния короткого замыкания и утечки тока по корпусу самолета [6].

Пропускная способность МКИО – 1 Мбит/с, допуск на погрешность и долговременный дрейф – 0,1%; краткосрочная стабильность тактовых импульсов  $\approx 0,01\%$ ; выходное напряжение передатчика – 18-27 В.

МКИО *Mil-Std-1553B* состоит из: двух шин (линий передачи информации: основной и резервной); контроллера шины; окончных устройств; монитора шины (рис. 1).

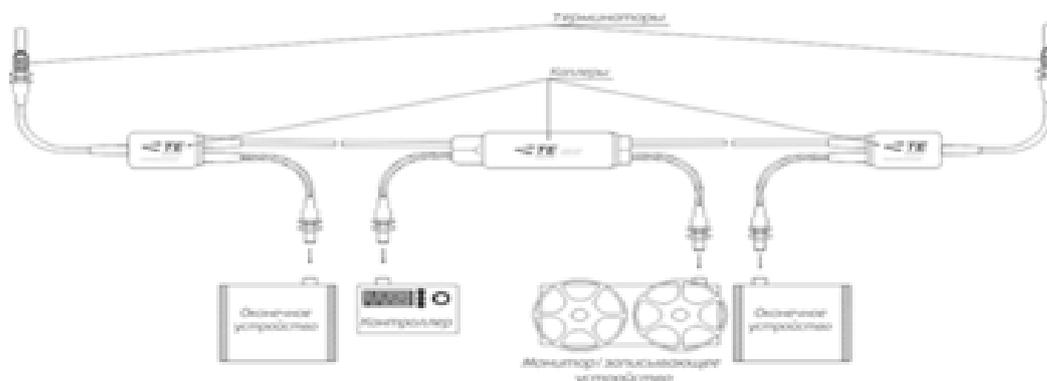


Рис. 1. Схема МКИО стандарта *Mil-Std-1553B*

На одном МКИО в любой момент времени может быть только один контроллер, который является инициатором всех сообщений в шине. Контроллер: оперирует командами; управляет передачей /приемом сообщений оконечными устройствами; обрабатывает запросы от оконечных устройств; фиксирует, восстанавливает и сохраняет ошибки и их историю. Оконечные устройства выполняют функции организации: взаимодействия шины данных и подключаемой подсистемы; моста между двумя и более МКИО. На одном МКИО может быть до 32 оконечных устройств. Монитор канала в отличие от оконечного устройства не может передавать сообщения по шине. Он выполняет мониторинг и запись

транзакцій (используется для последующего анализа) по шине данных, без вмешательства контроллера или оконечных устройств. На одном МКИО обычно используется не более двух мониторов.

### Модель бортового комплекса

Бортовой комплекс авионики – это многомашинная многоуровневая система (на рис. 2 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 -  $RT$  / *Remote Terminal*/, или как контроллер шины –  $BC$  / *Bus Controller*/ или как монитор шины –  $BM$  / *Bus Monitor*/. Отметим, что реально вместо ПК может быть штатная бортовая система технического обслуживания (БСТО) и/ или система бортовых измерений (СБИ) для этапа испытаний, то есть 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.

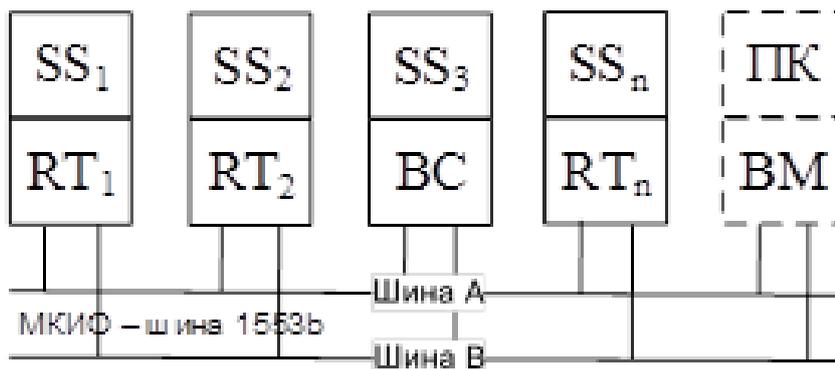


Рис. 2. Схема бортового комплекса авионики

Основные задачи интерфейса *Mil-Std-1553B* 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, которые позволяют моделировать информационные ситуации. Например, устройства *SM1553-1/3* (рис. 3) [7, 8] 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.



Рис. 3. Устройство *CM1553-1* и *CM1553-3*

### Устройство *CM1553-1/3*

Устройство *CM1553-1* – это плата *PCMCIA* типа III, обеспечивающая связь настольного/мобильного компьютера (ПК) с МКИО. Устройство совместно с мобильным ПК является мобильным программируемым пользовательским терминалом МКИО, который может выполнять функции *BC*, *RT* и *BM*, в том числе, на борту самолета.

Устройство *CM1553-1* реализует все функции обмена по МКИО, освобождая главный ПК от подготовки и обработки данных, что позволяет создавать адекватные информационные модели РВ в авионике.

Программное обеспечение (ПО) устройства *CM1553-1* – это драйвер окна *PCMCIA (Card Services)* и библиотека прикладных функций [7] для языка программирования высокого уровня. Например, это могут быть языки типа *C* и компиляторы *Borland C++/Turbo C*, *Microsoft C/Visual C++* [10]. В частности, программа *C11TEST* позволяет тестировать *CM1553-1* и проверить ее совместимость с ПО *Card Service*.

ПП управления *CM1553-1* включают от 1 до 7 элементов в зависимости от режима работы и требований прикладной задачи.

1. *Open* /открыть: *C11\_OpenCard* – первая функция в ПП, которая открывает доступ к *CM1553-1* посредством окна *PCMCIA* и дескриптора /*handle* (так как к ПК можно подключить несколько устройств).

2. *Configure* /конфигурировать: функция *C11\_<режим>Config* устанавливает режимы работы *CM1553-1*, где <режим>: *BC*, *RT*, *BM*, *RT/BM* (комбинация *RT* и *BM*). Возможны дополнительные установки. Например, в режиме *BC* объявляется таблица передаваемых сообщений /опции низкого уровня (работа в стандарте 1553A, разрешить прерывания).

3. *Initialize* /инициализировать: исключается передача /прием неправильных данных до перевода *CM1553-1* в активное состояние.

4. *Activate* /включить: перевод функцией *C11\_Start* *CM1553-1* в активное состояние, устройство взаимодействует с шиной данных без участия главного ПК (передает /принимает сообщения согласно конфигурации, без привлечения иных аппаратных /программных ресурсов).

5. *Handle Data* /управление данными: функции, получив доступ к данным, управляют потоками данных (ПК записывает в *CM1553-1* для передачи и считывает данные, получаемые по шине данных). Уменьшается загрузка главного ПК доступа к данным методом опроса /прерываний.

6. *Deactivate* /отключить: функция *C11\_Stop* переводит устройство из активного состояния в пассивное (устройство прекращает работу с шиной данных).

7. *Close* /закрыть: функция *C11\_CloseCard* прекращает работу ПП с устройством без перевода в пассивное состояние (задается в конце ПП).

Функции п.п. 3, 5 и 6 являются факультативными и зависят от прикладных задач.

### Решение задач

Основные задачи интерфейса *Mil-Std-1553B* (организация обмена информацией в МКИО между абонентами шины и управление этим обменом) решаются средствами программирования.

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*.

Рассмотрим несколько типовых базовых задач.

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

Алгоритм решения задачи передачи в прямом направлении (от *BC* 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  $n \neq WC$ ); *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.

## Р о з д і л 1 . І н ф о р м а ц і й н і с и с т е м и

Таким образом, *BC* 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 *BC*) 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, *BC* должен передать одно *CW*, после чего ждать заданное время ответа от *RT* в виде *SW* и необходимого числа *DW*. Соответствующий фрагмент ПП для *BC* 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 BC-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-BC>
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 *BC* и передачи данных между *BC* и *RT* в зависимости от бита *TR*; *BUSA /B* – определить обмен по шине *A* или *B*; *C11\_CloseCard (handle)* – функция отключения *CM1553-1*.

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

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

сообщение 2: *RT* с *AD=4* должен передать к *BC* *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: безосновательная загрузка ПК; возможные проблемы синхронизации. Лучший способ – это автоматическая синхронная передача фрейма с использованием отдельных функций. 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 BC 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 сообщении до их изменения функцией *C11\_BCWrData(\*)*. Последние принимаемые данные сохраняются и их можно прочитать функцией *C11\_BCRdData(\*)*.

**Задача 3.** Периодическая передача BC 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 *BC* 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, errmsg, 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  $f=1/time$  от *RT1* с *AD1* к *RT2* с *AD2*, то есть 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, *BC* должен передать 2 *CW*: первое в адрес *RT2* на прием информации, второе в адрес *RT1* на ее передачу, после чего ожидать от них ответов 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 *BM*: 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 МКИО в авионике:

- *BC*: 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.

- *BM*: совмещенный режим *RT* и *BM*.

### Применение

Понятно, что, имея большое количество устройств *СМ1553-1* и ПК, можно построить достаточно сложную адекватную реальной системе авионики моделирующую систему. Однако смысла в этом немного. Дело в том, что 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 (рис. 4).

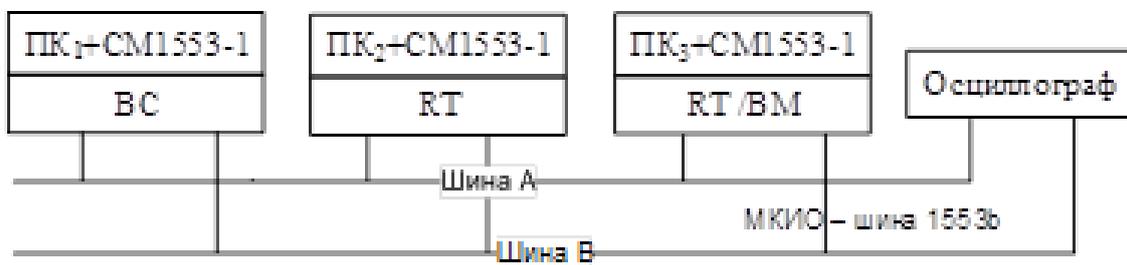


Рис. 4. Схема моделирующей системы авионики

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 [9].

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

## Р о з д і л 1 . І н ф о р м а ц і й н і с и с т е м и

/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-проверочная 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 LAC-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 (рис. 4) можно также использовать для обучения комплексированию и программированию интегральных комплексов авионики на базе интерфейса 1553В. В этом случае очень желательно ввести в систему двухлучевой осциллограф для наблюдения физических процессов в шинах МКИО. Одновременно могут работать три ученика как автономно, так и во взаимодействии в  $PB$  [3, 4, 11].

## Выводы

1. Применение в авионике цифровых технологий вызывает необходимость в подготовке специалистов, что, в свою очередь, требует наличия учебных пособий, преподавательских кадров и технической базы обучения. В частности, такая проблема существует в связи с внедрением стандарта *Mil-Std-1553B*.

2. Накопленный в Украине опыт применения интерфейсов 1553В в авионике позволяет организовать обучение для студентов по изучению принципов построения, взаимодействия и реализации компонентов авионики с применением программно управляемых связей.

3. Востребованность и внедрение информационных технологий в авионику, требует более широкого и существенного внимания к проблеме обучения. Опыт может только способствовать разрешению проблемы, на что авторы и надеются.

## Список использованной литературы

1. *MIL-STD-1553B*. Military Standard. Aircraft Internal Time Division Command/ Response Multiplex Data Bus. Notice 3. - Washington, DC 20360 USA, DODSTD, 1993. - 50p.

2. [What is MIL-STD-1553?](http://www.milstd1553.com) // <http://www.milstd1553.com>.
3. *Тихомиров В. О.* Проектування і експлуатація бортових комп'ютеризованих систем: Конспект лекцій. - К.: НАУ, 2002. - 124 с.
4. Проектування і експлуатація бортових комп'ютеризованих систем. Методичні вказівки і завдання на курсову роботу/ Уклад. В. О. Тихомиров. - К.: НАУ, 2002. - 52с.
5. *Хвоц С. Д., Амаду Х.Х.* Промышленные сети на базе стандарта MIL-STD-1553B // Современные технологии автоматизации. 1999. № 1/99. - С. 42–45. - // <http://www.cta.ru/cms/f/366607.pdf>
6. Компоненты шин информационного обмена по стандарту MIL-STD-1553B -<http://ptkgroup.ru/mil-std-1553b.html>
7. CM1553-1 User's Manual. Mil-Std-1553 PCMCIA Card and Software. - Everett, WA98201 USA, Ballard Technology, Inc., 1995. - 76 p.
8. Astronics Ballard Technology - Avionics Databus Solutions// <http://www.ballardtech.com/products.aspx/CM1553-1>.
9. RTCA/DO-178B. Software considerations in airborne systems and equipment certification. - Washington: RTCA, 1992. – 105 p.
10. *Зінченко В. П., Гераїмчук М. Д., Зінченко Н. П., ін.* Практичний курс програмування мовою С/С++. - К.: ІВЦ “Політехнік”, 2007. – 200 с.
11. *Зинченко В. П.* Концепция учебной лаборатории систем реального времени// Технології створення перспективних комп'ютерних засобів та систем з використанням новітньої елементної бази. – К.: НАН України Ін-т кібернетики ім. В. М. Глушкова, 2000. – С. 79 – 87.