This Software Programmers Manual (SPM) is written for the Generic Avionics Data Bus Tool Kit (GADBTK) Version 1.1. The GADBTK is an Ada Technology Insertion Program (ATIP) sponsored effort to produce an Ada binding with the MIL-STD-1553B time multiplex serial data bus.
1.0 SCOPE

1.1 Identification

This Software Programmers Manual (SPM) is written for the Generic Avionics Data Bus Tool Kit (GADBTK) Version 1.1. The GADBTK is an Ada Technology Insertion Program (ATIP) sponsored effort to produce an Ada binding with the MIL-STD-1553B time multiplex serial data bus.

1.2 System Overview

The GADBTK is to produce an Ada binding to the 1553B standard by defining the hardware elements and data structures used in a 1553 data bus system in terms of the Ada language. These definitions will then be used to provide various building block components from which 1553 data bus applications may be constructed. Finally, a bus monitor application will be constructed from the components as a proof of concept. Initial target system for this project will be a Digital Equipment Corporation VAXStation 3200 series computer with a Computer Technology & Simulation Dept developed Microprogrammable Multiplex Bus Interface (CTSD-MMBI).

1.3 Document Overview

This manual will detail the programming environment in which the 1553 interface was developed.
2.0 REFERENCED DOCUMENTS

2.1 Government Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

STANDARDS:
Military
MIL-STD-1553B, 8 September 1986 Digital Time Division Command/Response Multiplex Data Bus

OTHER:
SDD for the Generic Avionics Data Bus Tool Kit, 4 October 1991
SRS for the Generic Avionics Data Bus Tool Kit, 7 August 1991

2.2 Non-Government Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

STANDARDS:
AA-LA65A-TE, April 1988 VMS Programming Reference Set, Digital Equipment Corporation
3.0 SOFTWARE PROGRAMMING ENVIRONMENT

a. Equipment configuration.

In this application the host and target systems are the same. The system is a Digital Equipment Corporation VAXstation 3200 series CPU with a Trim Industries Q-BUS expansion chassis. Attached to the expansion chassis is a Bit-3 Corporation dual port shared memory. The other port of the dual ported memory connects to a Digital Equipment Corporation Uribus cabinet which is the host bus for the CTSD-MMBI adapter. The CTSD-MMBI adapter is a micro-programmable bus interface configured for 1553 bus communications.

b. Operational Information

1. Machine cycle time.
   n.a.

2. Word length
   The VAX word length is 32 bits.

3. Memory capacity and characteristics.
   The maximum memory is 64 megabytes.

4. Instruction set characteristics.
   See the VAX macro programming handbook.

5. Interrupt capabilities.
   There are 32 prioritized interrupt levels.

6. Modes of operation.
   The VAX supports batch, interactive, and network modes with four privilege levels in each mode.

7. Operational registers.
   See the VAX macro programming handbook.

8. Error indicators.
   See the VAX error handling facility manual.
9. Input / Output characteristics.
   See the VAX I/O Runtime reference manual.

10. Special features.
    n.a.

c. Compilations, assemblies, and linkages.
   The VAX VMS program development environment is based on an integrated set of
   editors, compilers, linker, and debugger. The linker and debugger are an integral part of the VMS
   operating system environment and are therefore tied in to the version of the operating system.
   This software was developed under version 5.4 of VAX VMS. Digital supplies several editors as
   a standard part of the operating system. In addition, they support a language sensitive editor
   called LSE which was used for the production of this program. The version of DEC Ada used for
   compilation was 2.3. To produce an executable image from the existing source code, create an
   Ada object library and load all source code files into the library. For information on how to do
   this please see the VAX Ada Programmers Reference Manual. After the source code has been
   loaded into the library, use the automated code management system compile command to compile
   the source code. Use the command "ACS LINK/DEBUG /NOOPT MONITOR_1553" to invoke
   the linker.
4.0 PROGRAMMING INFORMATION

a. Programming features.

1. Data representation.
   The VAX supports byte, word, longword, integer, floating, double precision floating,
   Ascii character, and packed decimal.

2. Instruction formats and addressing modes.
   See the VAX macro programming handbook.

3. Special registers and words.
   See the VAX hardware architecture manual.

4. Control instructions.
   See the VAX macro programming handbook.

5. Subroutines and procedures.
   See the VAX macro programming handbook.

6. Interrupt processing.
   See the VAX Device Driver manual.

7. Timers and clocks.
   See the VAX hardware architecture manual.

8. Memory protection features.
   See the VAX Device Driver manual.

9. Additional features.
   n.a.

b. Program instructions.
   n.a.

c. Input and output control programming.
1. Initial loading and verification of computer memory.
   n.a.

2. Serial and parallel data channels.
   RS 232 serial interface cards provide serial I/O. Parallel data for CTSD-MMBI interface is provided via DMA channel.

3. Discrete inputs and outputs.
   n.a.

4. Interface components.
   n.a.

5. Device numbers, operational codes, and memory locations for peripheral equipment.
   See the VAXstation hardware configuration manual.