INS/GPS Data Interface Using MIL-1553b Protocol in Airborne Radars

Amith S Patil, Arathi Pillai, D Aishwarya, Mrs. Priyadarshini M
Dept. of CSE, Cambridge Institute of Technology, Bangalore. India
Amith.12cs009@citech.edu.in, Arathi.12cs015@citech.edu.in, Aishwarya.12cs030@citech.edu.in, Priyadarshinim.cse@citech.edu.in

Abstract— ***INS/GPS data interface using MIL1553B protocol to detect the target through scanner from airborne radars where the navigation data (Navdata) is received from INS/GPS. The values longitude, latitude, azimuth and elevation of the target are given to the scanner to focus on the target through positioner. With respect to the value of longitude, latitude, azimuth and elevation (i.e., Navdata) the graph is plotted. Even the image of the target can be acquired. Taking data from INS/GPS navigation system every 20ms and commanding mechanical scanner for radar pointing operators.***

Keywords— INS/**GPS, MIL-STD-1553, scanner, radar.**

I. INTRODUCTION

MIL-STD-1553 is a military standard that defines the electrical and protocol characteristics for a data bus. A data bus is used to provide a medium for the exchange of data and information between various systems. It is similar to what the personal computer and office automation industry has dubbed a Local Area Network (LAN). [2] This guide provides an introduction to the MIL-STD-1553 data bus, its history, applications, and use. It describes:

- The physical elements that make up the bus.
- The protocol, including the message formats, word types, and command and status words.
- Status word bits and mode commands and their definitions and use, both from the remote terminal and bus controller perspective.
- Issues such as bus loading, major and minor frame timing, and error recovery.

II. LITERATURE SURVEY

- MIL-STD-1553, hardware specifications. Characteristics for a data bus. ARINC 629 limitations. [2]
- ARINC 429,708,629 AND IEEE 1394B existing systems specifications and history. Realistic implementations, the high amount of data bus traffic analysis.[1]
- ARINC 429 existing systems limitations, challenges and bandwidth analysis.[3]
- ARINC 708 existing systems limitations and signal deterioration. [4]
- IEEE 1394B existing systems limitations, isochronous channels and bandwidth analysis. [5]

III. EXISTING SYSTEMS

1. ARINC 429 :

"Digital Information Transfer System (DITS)," also known as Aeronautical Radio INC.(ARINC) is the technical standard for the predominant avionics data bus used on most higher end commercial and transport aircraft. It defines the physical and electrical interfaces of a two wire data bus and a data protocol to support an aircraft's avionics local Area Network. ARINC 429 is the most widely used data bus standard for aviation. Electrical and data format characteristics are defined for a two-wire serial bus with one transmitter and up to 20 receivers. The bus is capable of operating at a speed of 100 kbit/s. [1]

2. ARINC 708 :

ARINC 708 is a specification for airborne pulse Doppler weather radar systems primarily found on commercial aircraftman 708 uses a data transfer method using transformer coupled Manchester encoded signal, like the MIL-STD-1553 protocol. Termination of the bus is essential for good signal quality. Furthermore, the bit stream is continuous, requiring good resynchronization to the bit stream. Data frames are 1600 bits long with the header portion of the frame consisting of parameters such as range, tilt, gain, status, etc. The data portion is organized into 512 range-bits per scan angle value. Each range bit contains a color value to indicate the intensity at that position. Settings for the ARINC 708 system is typically controlled using an ARINC 429 interface. [1]

3. ARINC 629 :

ARINC 629 was introduced in May 1995 and is currently used on the Boeing 777, Airbus A330 and A340 aircraft. The ARINC 629 bus is a true data bus in that the bus operates as a multiple-source, multiple sink system. That is, each terminal can transmit data to, and receive data from, every other terminal on the data bus. This allows much more freedom in the exchange of data between units in the avionics system. The true data bus topology is much more flexible in that additional units can be fairly readily accepted physically on the data bus. A further attractive feature of ARINC 629 is the ability to accommodate up to a total of 128 terminals on a data bus shown in Figure, though in a realistic implementation the high amount of data bus traffic would probably preclude the use of this large number of terminals. It supports a data rate of 2 Mbps. [1]

4. IEEE 1394B :

IEEE 1394 is an interface standard for a serial bus for high-speed communications and isochronous real-time data transfer. It was developed in the late 1980s and early 1990s by Apple, which called it FireWire. The 1394 interface is comparable to USB though USB has more market share.[1] Apple first included FireWire in some of its 1999 Macintosh models, and most Apple Macintosh computers manufactured in the years 2000 - 2011 included FireWire ports. However, in 2011 Apple began replacing FireWire with the Thunderbolt interface and, as of 2014, FireWire has been replaced by Thunderbolt on new Macs.[2] The 1394 interface is also known by the brand i.LINK (Sony), and Lynx (Texas Instruments). IEEE 1394 replaced parallel SCSI in many applications, because of lower implementation costs and a simplified, more adaptable cabling system. The 1394 standard also defines a back plane interface, though this is not as widely used. [1]

IV. LIMITATIONS OF EXISTING SYSTEMS

1. ARINC 429

- Due to the simplistic layout of 429 links, each individual connection is a physical cable, which allows for easy testing, as either a LRU or the line itself may be faulty. This also poses a severe challenge when designing systems with dense interlinking.
- Even an environment with few stations present may become very complex once a certain degree of interaction is needed. In modern commercial aircraft, interlinking between a multitude of systems as well as extreme cabling overhead may occur, imposing severe limitations on network design as well as impacting the overall weight of such a craft.
- Bit-error correction via symmetric or cryptographic checksum algorithms is not designed to happen within ARINC 429 but instead needs to be implemented at application level. In fact, all data processing must be handled by each LRU's software directly. There is no uniform, device-independent 429 software stack.
- Overall, custom proprietary (and thus, expensive) hardware is required to implement an ARINC 429 setup, which is common in the aerospace industry. Almost no consumer-off the-shelf hardware is available, with the exception of cabling. However, it should be noted, separate aviation-standards apply to cabling. Software development is problematic too, as no modern day networking protocols can be used, and development is done for highly specialized hardware. Retrofitting of older aircraft with new technology may thus be costly.
- They are limited in bandwidth (100 kbps)
- They result in many wires because connections are mostly "point to point". [3]

2. ARINC 708

- ARINC 708 standard have cable lengths limitations like 6 (short stub) or 20 feet (long stub).
- Several devices are connected to a single stub, which can create signal deterioration such that only 10% of the initial transmitted signal is sent to the receiver side. [4]

3. ARINC 629

- **Expensive**: The equipment required for this protocol is expensive compared to other protocols and hence requires more maintenance.
- **Bandwidth** is technically limited and shared. When the bandwidth is shared there are chances for collisions and traffic or the signal losing its strength. [2]

4. IEEE 1394B

- **Number of Isochronous channels:** Although the 1394a standard limits the maximum number of simultaneous isochronous channels to 16, there is currently no host adapter that is capable of supporting 16 channels. Host adapters based on the TI chipset can support at most 4 simultaneous DMA channels (or contexts).
- **Maximum IEEE 1394 Bandwidth:** The maximum bandwidth of the 1394a bus is 400Mb/s, or 50MB/s. This bandwidth must be split between isochronous (e.g. image streaming) and asynchronous (e.g. camera control) data transfers. 80% (40MB/s) of the bandwidth is allocated to ISO transfers and the other 20% (10MB/s) to ASYNC. The maximum size of a single data packet, as defined by the 1394a Trade Association and enforced by the Microsoft Windows 1394 driver stack (1394bus.sys, ohci1394.sys, etc.) is 4096 bytes. With 8000 ISO cycles per second, this equals approximately 33MB/s, which means there is an additional 7MB/s available for ISO data. Therefore, it is possible to allocate up to 4915Bytes/packet, although this violates the 1394a specification.
- **Maximum Number of Devices**: The 1394a standard limits the maximum number of devices on a single bus to 63. [5]

V. DESIGN

A. Context diagram

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, organizational groups, external data stores, etc.). Benefits of a Context Diagram are:

- Shows the scope and boundaries of a system at a glance including the other systems that interface with it

- No technical knowledge is assumed or required to understand the diagram
- Easy to draw and amend due to its limited notation
- Easy to expand by adding different levels of DFDs
- Can benefit a wide audience including stakeholders, business analyst, data analysts, developers.

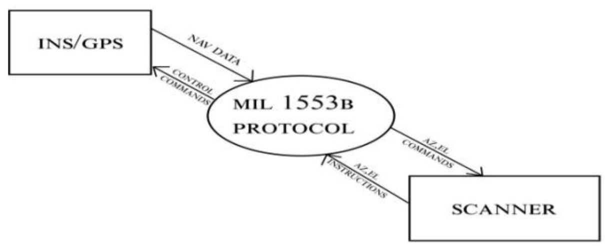


Fig. 1. Context diagram of INS/GPS data interface over MIL 1553B protocol in airborne radar applications

B. Architectural design

The architectural design we are using here is based on object model diagram. The object model diagram is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time. The figure below shows the architectural design of the system.

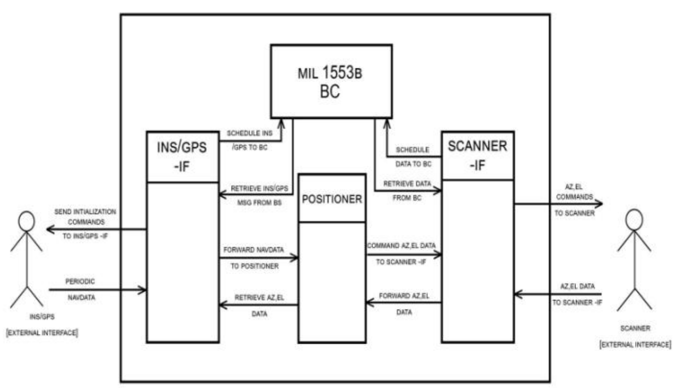


Fig. 2. Archetectural design(object model diagram)

As seen in the above diagram (Fig 2), there are 4 main components-

- INS/GPS_IF (INS/GPS interface)
- Positioner
- Scanner_IF(Scanner interface)
- MIL 1553B BC (bus controller)

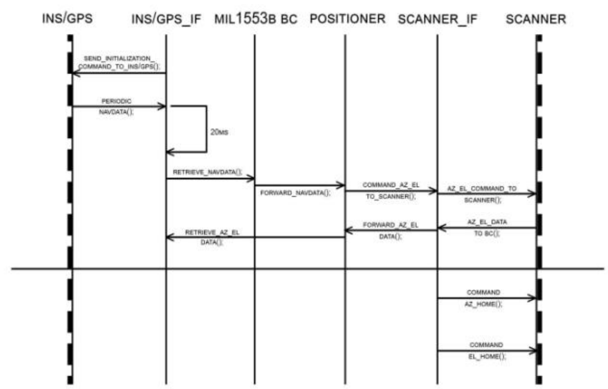
There are two external interfaces, ie, INS/GPS and Scanner. The INS/GPS and Scanner are responsible for sending and receiving commands across the MIL 1553B bus. The INS/GPS will send periodic Navdata to INS/GPS_IF which then schedules the INS/GPS message to BC. The BC then services this message and sends it to the Scanner_IF which then sends to the Scanner as AzEl commands.

The Scanner will send the Az-El data to Scanner_IF which schedules the data to BC which services this and sends the INS/GPS message to INS/GPS_IF which sends initialized commands to INS/GPS. The Navdata from INS/GPS_IF can be forwarded to the Positioner which processes it and sends Az-El commands to Scanner_IF and vice versa.

C. Sequence diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

In the below sequence diagram, shows how the system works. The INS/GPS and Scanner are the external interfaces. There is INS/GPS_IF, MIL 1553B BC, Positioner and Scanner_IF. The INS/GPS will send periodic Navdata to INS/GPS_IF which then schedules the INS/GPS message to BC.



The BC then services this message and sends it to the Scanner_IF which then sends to the Scanner as Az-El commands.

The Scanner will send the Az-El data to Scanner_IF which schedules the data to BC which services this and sends the INS/GPS message to INS/GPS_IF which sends initialized commands to INS/GPS. The Navdata from INS/GPS_IF can be forwarded to the Positioner which processes it and sends Az-El commands to Scannner_IF and vice versa. Another action is when the Scanner_IF sends commands to Scanner making it move at 0 degree with AzHome() and ElHome(), mostly to reset the device.

CONCLUSION

This is the precursor to many of the protocols used today. And yet, despite its age, it’s still used whenever reliable data communications under severe environmental and EMI conditions are absolutely necessary. This report is far from exhaustive, but dwelling on the operational details would’ve made its size unmanageable and obscured the engineering concepts you’ll need to understand in order to appreciate the new protocols that are currently being introduced into service.

This system will help the flight test engineers and pilots to find out the position of the target aircraft and display the range and bearing of the target aircraft. It could be used by the test pilots during the on-line mode. During the offline modes the pilots' trajectory, altitude, speed, longitude and latitude can be traced.

References

[1] E. M. Atkins, R. H. Miller, T. VanPelt, K. D. Shaw, W.B. Ribbens, P. D. Washabaugh, and D. S. Bernstein, ''Solus: An Autonomous Aircraft for Flight Control and Trajectory Planning Research,'' Proc. of the American Control Conference, pp. 689- 693, 1998.

[2] Truitt, R.B. Sanchez, Eugaries', M. Autotestcon 2004. Proceedings Volume, Issue, 20- 23 Sept. 2004 Page(s): 117 -- 123 ''Using open networking standards over MIL-STD-1553 networks''

[3] The Evolution of Avionics Networks From ARINC 429 to AFDX ChristianM.Fuchs

[4] http://bmccorp.com/PAGE24.asp

[5] https://www.ptgrey.com/kb/10066