

ARINC 818-2 Protocol Overview

ARINC 818 is the **Avionics Digital Video Bus (ADVB)**: a fiber-channel-based, point-to-point serial link for uncompressed, high-bandwidth, low-latency video and data in aircraft ¹ ². It was developed by Airbus, Boeing and avionics suppliers for modern cockpits (787, A400M, A350, etc.) that demand high-resolution video for displays, sensors and HUDs ¹ ³. The original ARINC 818 (2006) was based on Fibre Channel Audio/Video (FC-AV) but **simplified** (no handshaking, flow control, etc.), retaining only high speed (up to 8.5 Gb/s), reliability and precise display timing ¹ ². Video is framed into *containers* (each carrying one video image) which are split into fixed-size *ADVB frames* (up to 2112 bytes of payload) for transmission. Each container and frame include headers with frame numbers, source/destination IDs, and a 32-bit CRC ⁴ ².

ARINC 818-2 (published Dec 2013) is the *supplement* that formally adds many features and higher data rates that had emerged in practice. It defines **new fiber-channel link rates** (up to 28.05 Gb/s and beyond), plus optional support for field-sequential color, data-only return channels, channel bonding (parallel links), bi-directional links, and flags for encryption/compression ⁵ ⁶. In essence, ARINC 818-2 is a robust “video transport protocol” optimized for avionics: it carries RGB or grayscale pixel streams (progressive or interlaced) along with ancillary audio or control data, all with strict timing. Its advantages include **very low latency** (often well under one video frame) ⁷, deterministic behavior (synchronous pixel clocks), and two layers of error checking (running disparity + CRC) for mission-critical safety. It is used in military and commercial aircraft worldwide – for example as the display link on the Boeing 787, Airbus A350/A400M, KC-46 tanker, C-17 and C-130J, F-15/F-18 upgrades, etc. ⁸ ⁹.

Key Differences from ARINC 818-1

ARINC 818-2 **extends** the 2006 standard. The new features include:

- **Higher Link Rates:** Supplement 2 formally adds Fibre Channel 6x, 12x, 16x, 24x, 32x and other intermediate rates (e.g. 5.0 Gb/s, 6.375 Gb/s, 12.75 Gb/s, 21.0375 Gb/s, 28.05 Gb/s) ¹⁰ ⁶. These cover link rates that many new FPGAs and hardware now support.
- **Channel Bonding:** Multiple physical links in parallel can carry one video frame (splitting the pixel data across links). This allows, for example, a 2560×1600 @60 Hz image split across two 4.25 Gb/s links if no single link is fast enough ¹¹.
- **Field-Sequential Color:** A new Video Format Code (VFC) was added for field-sequential mode. In this mode each primary color (and possibly others) is sent in separate containers. (For example, one container with all-red pixels, then one with green, etc.) This can reduce hardware cost for certain displays ¹².
- **Data-Only Return Paths:** ARINC 818-2 explicitly supports a second, bidirectional link for camera/sensor control or touchscreen return data. Such links carry arbitrary data frames (no video) for functions like focus adjustment or touch input ¹³.
- **Compression/Encryption Flags:** The spec adds optional container flags indicating whether the payload is compressed or encrypted, so heterogeneous links can optionally carry compressed sensor data ¹⁴. (Actual codec or cipher is left to the system's ICD.)

- **Switching Guidelines:** ARINC 818-1 was strictly point-to-point. Supplement 2 formalizes some basic rules for switches/routers: e.g. only frame-by-frame switching (no mid-frame cuts) to avoid tearing, since containers have source/dest IDs ¹⁵ .
- **Other Features:** Support for stereo/3D displays, optical signal quality specs, prior-image CRC, etc. ⁶ .

In short, ARINC 818-2 brought ARINC 818 into line with newer hardware speeds and added flexibility (compression, switching, bidirectional, etc.), while still maintaining its lightweight, deterministic packet format. All ARINC 818-2 implementations use an **Interface Control Document (ICD)** to fix parameters (resolution, pixel format, frame rate, sync mode, etc.) so that transmitter and receiver agree exactly ¹⁶ .

Protocol Layers and FPGA Logic

ARINC 818 is essentially a **serial link protocol** layered on a Fiber Channel physical link. The layers can be thought of as:

- **Physical/Link Layer:** Uses a Xilinx transceiver (GTX/GTH/GTP/GTZ, etc.) configured for 8b/10b (for rates up to 28.05 Gb/s) or 64b/66b encoding (for higher rates in ARINC 818-3). The transceiver serializes 8-bit bytes into 10-bit symbols, adds idle characters during blanking, and handles *ordered sets* for Start-of-Frame (SOF) and End-of-Frame (EOF) markers ¹⁷ ¹⁸ . The transceiver has built-in block coding, clock/data recovery, and running-disparity handling. Xilinx FPGAs provide *Gigabit Transceiver* (GT) primitives for these roles. The FPGA logic drives the transceiver's TX interface with 16-bit (or 32-bit) words per cycle, and monitors its RX interface for frame markers, comma alignment, disparity errors, etc.
- **ADVB Frame Layer:** Data is packetized into *ADVB frames*. Each ADVB frame begins with an SOF ordered set (to mark frame start) and ends with an EOFx ordered set (EOF Normal or EOF Terminate) ¹⁸ . Between these markers, each frame carries a 6-word header plus up to 2112 bytes of payload. The header includes the source/destination IDs and a **frame index** (position within the container) ⁴ ² . The payload is either Ancillary data (Object 0) or Video pixel data (Object 2) as defined in the container profile. (Object 1 and 3 are for audio/odd-field in interlaced modes ⁴ .) All frames have a 4-byte CRC appended after the payload, computed with the FC polynomial ¹⁹ ²⁰ . If a video line exceeds 2112 bytes, it is split across multiple ADVB frames (as in XGA example where 3072 bytes of line data become two frames) ⁴ .

Figure: ARINC 818 framing. A video frame is split into an ADVB "container" which is then sent as a sequence of ADVB frames. The first frame carries the container header (ancillary data) and later frames carry video pixels. Each ADVB frame ends with an EOF ordered set and a CRC word ⁴ ²¹ .

- **Container/Video Layer:** A *container* spans many ADVB frames and carries one complete video image. The container header (Object 0) includes the video format code (timing class, resolution, pixel format) and timestamp info. Video pixel data (Object 2) fills subsequent frames line by line. The FPGA must implement a **frame generator state machine** that, on each video-vertical-sync, begins pulling pixel words (e.g. RGB or YCbCr pixels) from line buffers and assembling them into ADVB frames. The first ADVB frame of the container is usually shorter (it carries header+ancillary); following frames

carry only video data ⁴ ²² . At the end of the frame (vertical blank), a final EOFt is sent and the 32-bit CRC is output.

In FPGA logic, the “*link layer*” and “*transport layer*” duties map directly to HDL modules. A typical design has: video input blocks (e.g. from a GPU or camera), line/frame buffers (BRAMs or FIFOs), a frame-builder FSM, a CRC engine, and an interface to the multi-gigabit transceiver. On the RX side, logic must reassemble containers from received frames, check CRCs, extract pixel data, and resynchronize video clocks. Many of these functions can be written in VHDL/Verilog or obtained as IP cores. For example, Xilinx Virtex/Kintex families provide built-in 8b/10b PCS blocks in the GTX/GTH (UltraScale) transceivers, and great river or third-party IP cores supply the ADVB formatting logic ²³ ²⁴ .

Data Formats: Resolutions and Pixel Encoding

ARINC 818-2 supports essentially **any uncompressed pixel format** defined by the ICD. Common modes include 8-bit or 10-bit **RGB** and **YCbCr** color, grayscale, or RGBA pixels ²⁵ . (The iWave IP core, for instance, explicitly supports Monochrome, RGB, YCbCr, and RGBA pixel formats ²⁵ .) Video may be progressive or interlaced. Pixel aspect ratios (e.g. 1:1, 4:3 (NTSC), 16:15 (PAL), etc.) are also accommodated.

Typical display resolutions in avionics range from VGA/XGA through 1080p and up to 4K. For example, 1080p60 (1920×1080 @60 Hz, 24-bit) requires about 3.0 Gb/s of payload ²⁶ . The table below illustrates some common data rates (24-bit color, no compression) and the corresponding ARINC 818 link:

- **XGA (1024×768 @60 Hz):** $\approx 1.13 \text{ Gb/s} \Rightarrow \text{FC } 2\text{x} (2.125 \text{ Gb/s})$ ²⁷ .
- **Full HD (1920×1080 @60 Hz):** $\approx 2.99 \text{ Gb/s} \Rightarrow \text{FC } 4\text{x} (4.25 \text{ Gb/s})$ ²⁷ .
- **4K UHD (3840×2160 @30 Hz):** $\approx 5.97 \text{ Gb/s} \Rightarrow \text{FC } 6\text{x} (6.375 \text{ Gb/s})$ ²⁸ .
- **4K DCI (4096×2160 @30 Hz):** $\approx 6.37 \text{ Gb/s} \Rightarrow \text{FC } 8\text{x} (8.5 \text{ Gb/s})$ ²⁹ .

These examples (from a Great River design guide) show that uncompressed 1080p60 already pushes a single ARINC 818 link near 3 Gb/s, so higher resolutions often use higher fiber-channel rates or multiple links ³⁰ . Pixel word size is typically 8 bits per color (some systems use 10-bit); note that 8b/10b encoding adds 20% overhead on the link.

In summary, ARINC 818-2 can carry any rectangular video (up to arbitrarily large resolutions at lower frame rates, or moderate resolutions at high frame rate), and the exact format (RGB vs YCbCr, bit depth, scan type, etc.) is negotiated in the system ICD. This flexibility means ARINC 818 links have been used for everything from simple head-down displays (1280×1024) to very high-res helmet displays, sensor cameras (infrared SWIR, etc.), and more ³¹ ⁹ .

Xilinx FPGA Implementation

Implementing ARINC 818-2 on a Xilinx FPGA typically involves a mix of vendor IP cores and custom HDL. Key elements are:

- **IP Cores:** Companies like iWave, Techway and Great River sell ARINC 818 IP cores (VHDL/Verilog) that handle the protocol framing. For instance, an iWave core can be instantiated on any transceiver-capable FPGA and provides a streaming video interface to your logic ²³ . These cores handle the

ADVB frame assembly, CRC, and protocol compliance. Using an IP core saves development time, but designs can also be hand-coded if desired (see Great River's **Make vs Buy** white paper ³²).

- **Multi-Gigabit Transceivers (MGTS):** A Xilinx FPGA's GTX/GTH/GTP transceivers form the physical link. They are configured for the desired rate (8b/10b or 64b/66b) and encode/decode the serial data. In HDL, you feed each transceiver a parallel word each clock: typically 16- or 32-bit at DDR. One must also handle transceiver signals like **RX_LOSS** (recover sync lost), **DISP_ERR** (8b/10b decode errors), and running-disparity status. Note that transceiver pinout and status signals differ by FPGA family: e.g. Virtex-5 vs Kintex-7 vs UltraScale have different interfaces ³³ . Porting designs between Xilinx families (or to/from Altera) may require adapting to these differences ³⁴ . In practice, Xilinx's Transceiver Wizard IP or IBERT core is used to configure/generate the MGT settings and perform bit-error-rate (BER)/eye tests during bring-up.
- **Frame Builder and CRC:** The FPGA needs logic (usually a small FSM) that, on each video frame, reads pixel data (from DMA or video input logic), assembles the container header and video payload into a stream, and signals the transceiver's TX interface. A 32-bit CRC generator computes the ADVB CRC across each container ¹⁹ . The first ADVB frame is assembled when a vertical-sync occurs: it includes header (Object 0) and possibly padding; subsequent frames carry pixel data (Object 2) until the end of frame. On the RX side, logic strips idle words, detects SOF/EOF ordered sets, validates CRC, and reassembles pixels. (All of this can be built from FIFOs, comparators, and a CRC checker in logic.)
- **Buffering and Clock Domains:** Typically the video source (pixel clock) is in a different clock domain from the high-speed serial. Dual-port BRAMs or FIFOs are used to cross these domains. For example, a line buffer can accumulate one video line at pixel rate, then the frame FSM reads it out at the link clock rate. Xilinx tools provide CDC-aware FIFOs/BRAMs. The recovered pixel clock from the transceiver may not align with system clocks, so care must be taken to resynchronize or use elastic buffers. Great River notes that clocking schemes and resets are a key challenge – for instance, holding the MGT in reset until a valid signal appears ³⁵ . Also, because ARINC 818 is unidirectional, one side recovers the clock from the incoming data stream, which must then be used for local pixel timing.
- **GPIO and Ancillary Data:** Often, ARINC 818 links carry small amounts of auxiliary data. IP cores usually provide “blanking-period” channels or packet fields where user data can be inserted/received (e.g. menu commands, timestamps, audio). The FPGA design must multiplex this data into the appropriate parts of the ADVB frames.

In Xilinx terms, the **overall block design** might include: video source (AXI4-Stream or custom video IP) → pixel formatter → **ARINC 818 TX core** (could be custom HDL or 3rd-party) → Transceiver TX. On receive: Transceiver RX → **ARINC 818 RX core** → pixel output. The Vivado IP catalog doesn't include ARINC 818 by default, so designers use external IP or their own code. The transceiver interface (the PHY layer) is typically built using Xilinx's GT wizard or subsystem, which handles PLL resets, CDR calibration, 8b/10b coding, etc.

Timing, Latency, and Bandwidth

ARINC 818 was engineered for *real-time avionics video*, so **latency** is minimized. Because the video is uncompressed, the main delay is buffer/frame assembly. In good designs, end-to-end latency can be well under one frame interval. For example, a ping-pong buffer scheme yields ~1 frame of latency (16 ms at 60 Hz) ³⁶, and streaming FIFOs can achieve sub-frame latency. This is critical for HUDs or sensor fusion, where higher latency would misalign graphics and pilot view ³⁶. There is no hard limit on ARINC 818 frame rate – the spec will carry any frame rate as fast as the link allows (e.g. a very high frame-rate ROI scan).

Bandwidth planning is crucial. The required link bit-rate = (pixels×bits_per_pixel×frames_per_sec) + overhead. Xilinx designers must ensure the FPGA's MGT can sustain the chosen rate (e.g. 12.75 Gb/s or higher on UltraScale GTYs). The 8b/10b encoding adds 20% overhead, and each ADVB frame has ~32 bytes of header+CRC overhead per 2112 bytes of payload. These overheads must be included in timing budgets.

For example, a 30-bit RGB color (24-bit RGB + 6-bit sync) video at 60 Hz (1920×1080) yields ~3.6 Gb/s including overhead; this needs at least a 4.25 Gb/s link. More aggressively, a 4K×2160 @60 Hz (24-bit color) is about 12.0 Gb/s of payload, requiring a 14.025 Gb/s link or two bonded links. Because of this, ARINC 818-2 added link rates up to 28.05 Gb/s for very high-resolution or high-framerate systems ¹⁰ ³⁷. Designers must check timing across the FPGA fabric: at multi-Gb/s, PCB trace lengths, logic pipelining, and clock skew all become critical.

Great River's guide also emphasizes built-in **error monitoring**: the FPGA should count running-disparity errors, CRC failures, and loss-of-lock events for health monitoring. Such diagnostics are often fed into the aircraft's maintenance bus (e.g. ARINC 664/AFDX data).

Tools and Debugging

In practice, Xilinx Vivado (or ISE for older devices) is used for ARINC 818 FPGA designs. **Integrated Logic Analyzer (ILA)** cores are invaluable for probing internal signals (state machines, buffers) in real hardware. Designers also rely on Xilinx's **Integrated Bit Error Ratio Test (IBERT)** core to test transceiver links: IBERT can sweep voltage/offsets to visualize a 2D eye diagram and measure BER on the ARINC 818 link. The GT wizard in Vivado provides loopback modes and eye scan features for GTX/GTH blocks. Debugging often involves JTAG probing of the ILA to trigger on ADVB frame events, and using board hardware (oscilloscopes or SerDes analyzers) to verify 8b/10b symbols.

If using a Xilinx SoC (e.g. Zynq) one can even run software to feed video frames via DMA and then read them back, which helps validate the whole link. Third-party test equipment exists (e.g. ARINC 818 analyzers, LinkTesters) but many labs just emulate ARINC 818 signals with FPGAs. Vivado's simulation tools (e.g. ModelSim) can simulate ARINC 818 timing classes if provided with a testbench.

Aerospace Applications Examples

ARINC 818 (ADVB) is widely deployed in modern aircraft and sensor systems. Some examples of usage:

- **Cockpit Displays:** It is the *primary digital video link* to glass cockpits. Boeing 787 Dreamliner, Airbus A350/A400M, and upgraded fighters like the F-18E/F use ARINC 818 to drive their multifunction displays ⁸ ². The protocol's determinism and low latency are especially valuable for head-up displays and synthetic vision overlays.
- **Sensors and Cameras:** High-resolution IR or CCD cameras on jets use ARINC 818 to send raw video to processors ³¹ ³⁸. For example, an infrared camera pod on a UAV could stream 4K60 via a 12.75 Gb/s ARINC link. The bi-directional feature is used for camera control (focus, zoom) as a return channel ³⁹. ARINC 818's adoption in missile seekers and surveillance radars is increasing (as noted by Great River and Airbus/Honeywell projects).
- **Video Concentrators and Buses:** Some systems multiplex or switch ARINC 818 streams. "Video switches" built on FPGAs route multiple ARINC 818 inputs to outputs (only allowed to switch on vertical blank) ¹⁵. Data concentrators merge multiple sensor streams onto one fiber using source/dest IDs. This is analogous to "networking" of video in the aircraft.
- **Commercial Interiors:** While not avionics-certified, ARINC 818-2 can also carry in-flight entertainment or passenger-camera data. One press report notes seat-back displays and phones getting their video from an ARINC 818-2 backbone, enabling features like on-board tablets with cockpit-like video feeds ⁴⁰.
- **Avionics Products:** Companies like Rockwell Collins and Thales ship products (e.g. the ProLine Fusion avionics suite, TopDeck helmet-display) that use ARINC 818 inside. Great River and iWave modules provide ARINC 818 interfacing for cockpit systems. An example project cited by iWave implemented ARINC 818-2 on a next-gen fighter (by India's ADA) using a Xilinx FPGA ⁴¹.

In summary, ARINC 818-2 is now the de-facto **video interface standard** for high-performance avionics and sensors. Its use spans from the largest airliner cockpits to military sensor pods and helmet displays, wherever uncompressed, low-latency video over fiber or coax is needed ⁹ ³⁸.

Sources: ARINC 818-2 specification literature and FPGA application notes ¹ ⁶ ⁴² ⁴ ²³ ³⁰ ³³ (citations indicate source pages and line ranges).

¹ ⁵ ⁸ ¹⁰ ¹¹ ¹² ¹³ ¹⁴ ¹⁵ Overview of ARINC 818 Supplement 2 - Avionics International

<https://www.aviationtoday.com/2014/04/07/overview-of-arinc-818-supplement-2/>

² ³ ²⁰ Explaining ARINC 818 - Avionics International

<https://www.aviationtoday.com/2008/03/01/explaining-arinc-818/>

⁴ ¹⁶ ¹⁷ ¹⁸ ¹⁹ ²¹ ⁴² How Does ARINC 818 Work? | ARINC818.com

<https://www.arinc818.com/how-does-arinc-818-work/>

⁶ ⁷ ³¹ ³⁶ arinc818.com

<https://www.arinc818.com/wp-content/uploads/2017/04/arinc-818-NewInSupp2.pdf>

⁹ ²⁶ ²⁷ ²⁸ ²⁹ ³⁰ ³⁷ Microsoft Word - Implementers2018

https://docs.wixstatic.com/ugd/ae08_37d298772ed9485f97af8f2954270e38.pdf?index=true

22 24 32 33 34 35 **IMPLEMENT ARINC 818 | Great River Technology**

<https://www.greatrivertech.com/implementarinc818>

23 25 41 **ARINC 818-2 IP Core - iWave Systems**

<https://www.iwavesystems.com/product/arinc818-2-ip-core/>

38 39 **SENSORS ON ARINC 818-2 | Great River Technology**

<https://www.greatrivertech.com/arinc818sensors>

40 **ARINC 818-2: Unveiling Characteristics and Boundless Applications**

<https://arincinsider.com/arinc-818-2-unveiling-characteristics-and-boundless-applications/>