Test-as-You Fly SpaceWire for Solar Probe Plus

Rich Conde, SPP Deputy Spacecraft System Engineer
Andrew Harris, SPP Testbed Software Lead
Alan Mick, SPP Data System Engineer

This presentation does not contain ITAR restricted information.

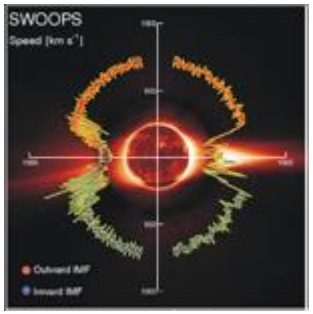
The Johns Hopkins University
APPLIED PHYSICS LABORATORY

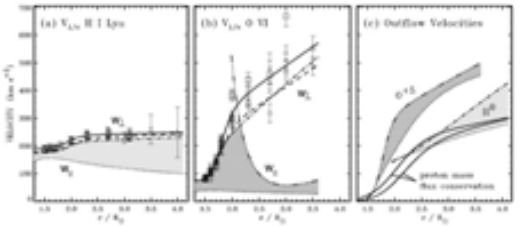
Summary

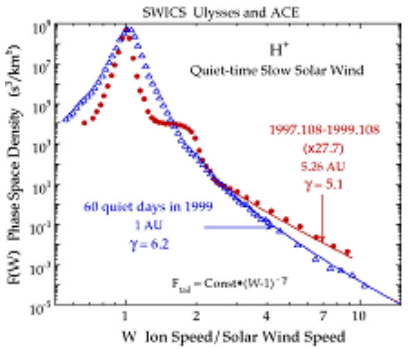
- **The Solar Probe Plus uses SpaceWire as the primary data interconnect between the Single Board Computers, the Avionics, and the Radios.**
- **SpaceWire, specifically the SPP router, has capabilities that can greatly simplify spacecraft test configurations, simplify simulation of missing spacecraft components, and allow injection of data for testing fault conditions.**
- **These capabilities are:**
  - **Logical addressing**
  - **Packet duplication**
  - **Remote updating of the routing table**

Solar Probe Plus Science Objectives

- **Primary Science Goal:**
  - **Determine structure and dynamics of Sun's coronal magnetic field, understand how the solar corona and wind are heated and accelerated, determine what mechanisms accelerate and transport energetic particles**
- **The primary SPP mission science goal defines three overarching science objectives**
  - **Trace the flow of energy that heats and accelerates the solar corona and solar wind**
  - **Determine the structure and dynamics of the plasma and magnetic fields at the sources of the solar wind**
  - **Explore mechanisms that accelerate and transport energetic particles**



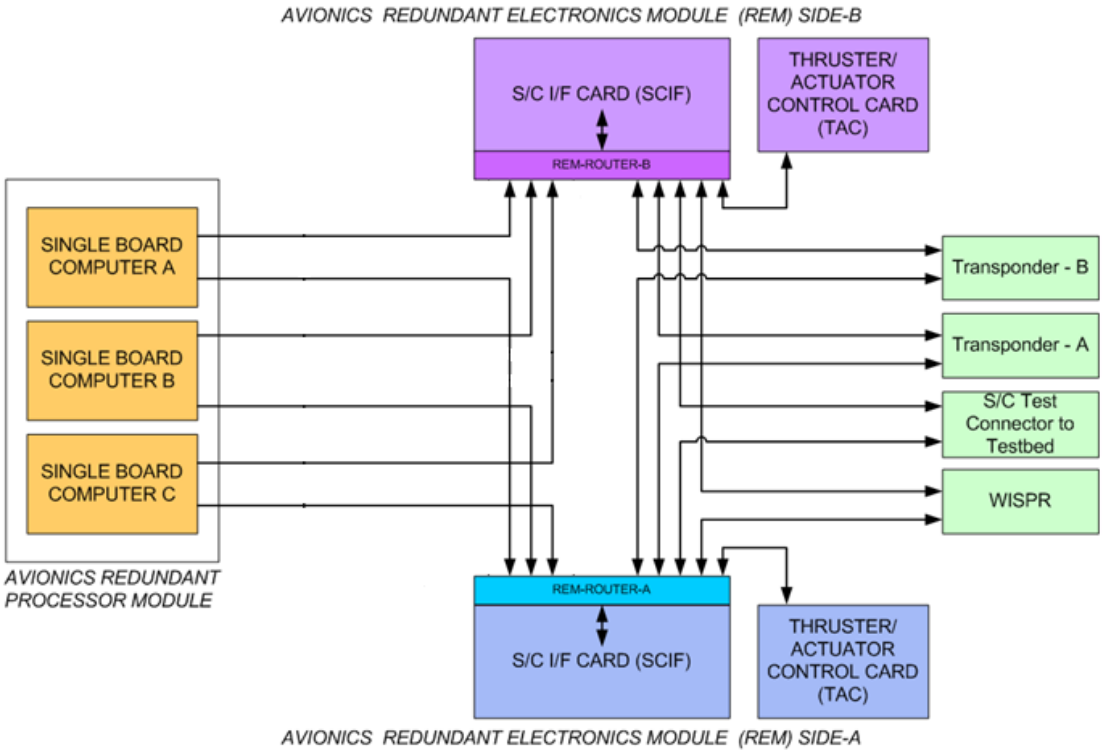




Spacecraft Overview

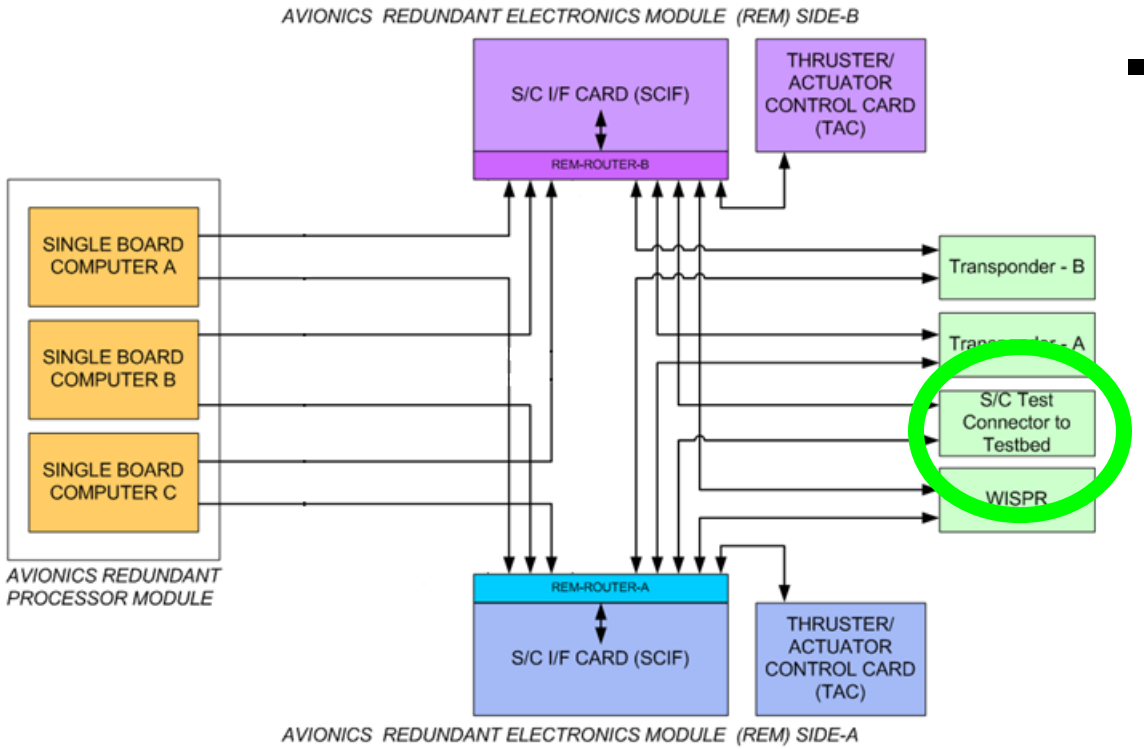
- Three Axis Stabilized
  - Wheels + Thrusters for Momentum Dumping.
- Ceramic Coated Carbon-Carbon Thermal Protection Shield (TPS)
- Actively Cooled Solar Power System
  - Water Cooled Solar Array Substrates
  - Mechanical Pump Loop
  - Radiator area under TPS
- Design Drivers:
  - Solar Environment
  - Mass
  - Power

Solar Probe Avionics and SpaceWire Network



- **SpaceWire Selected over 1553:**
  - **Greater Bandwidth**
  - **Lower Emissions**
- **Redundant Processor Module**
  - **Prime, Hot Spare, Backup Spare**
- **Redundant Electronics Modules**
- **Two Cross Strapped Transponders**
- **Cross Strapped Imager**

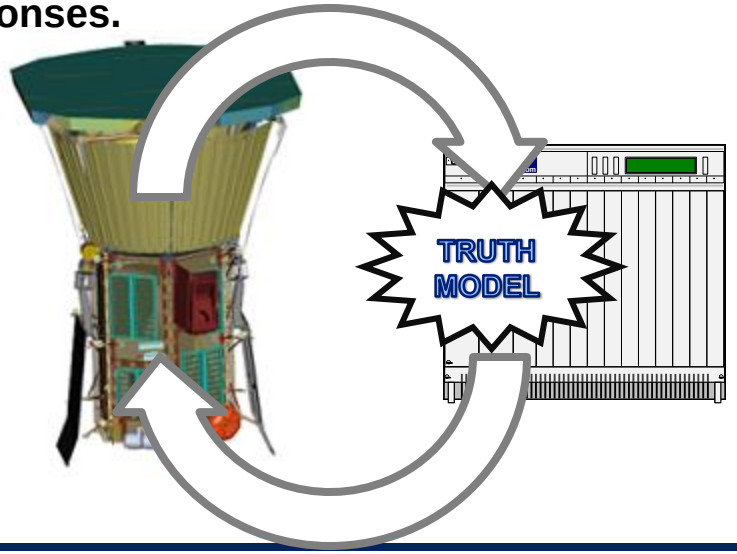
Solar Probe Avionics and SpaceWire Network



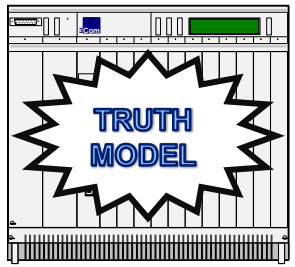
- **NOTE: SpaceWire Router provides connection the Spacecraft testbed.**

Testbed Functional Responsibilities

- **The testbed is responsible for the following capabilities:**
  - **The testbed substitutes for components that are under development or that have been "swapped out" for various reasons during integration testing.**
  - **The testbed maintains a "truth model" that simulates the dynamics of the spacecraft based on the history of reaction wheel / thruster actuation.**
  - **The testbed updates avionics telemetry with the "truth" maintained by the truth model.**
  - **The testbed allows injection of non-nominal data to simulate fault conditions for testing autonomy responses.**



Tried and True Implementation (Notional)



Flight Software CPU

SpW

SpW

Avionics SIDE A

A CMD
A TLM
B CMD
B TLM

Avionics SIDE B

A CMD
A TLM
B CMD
B TLM

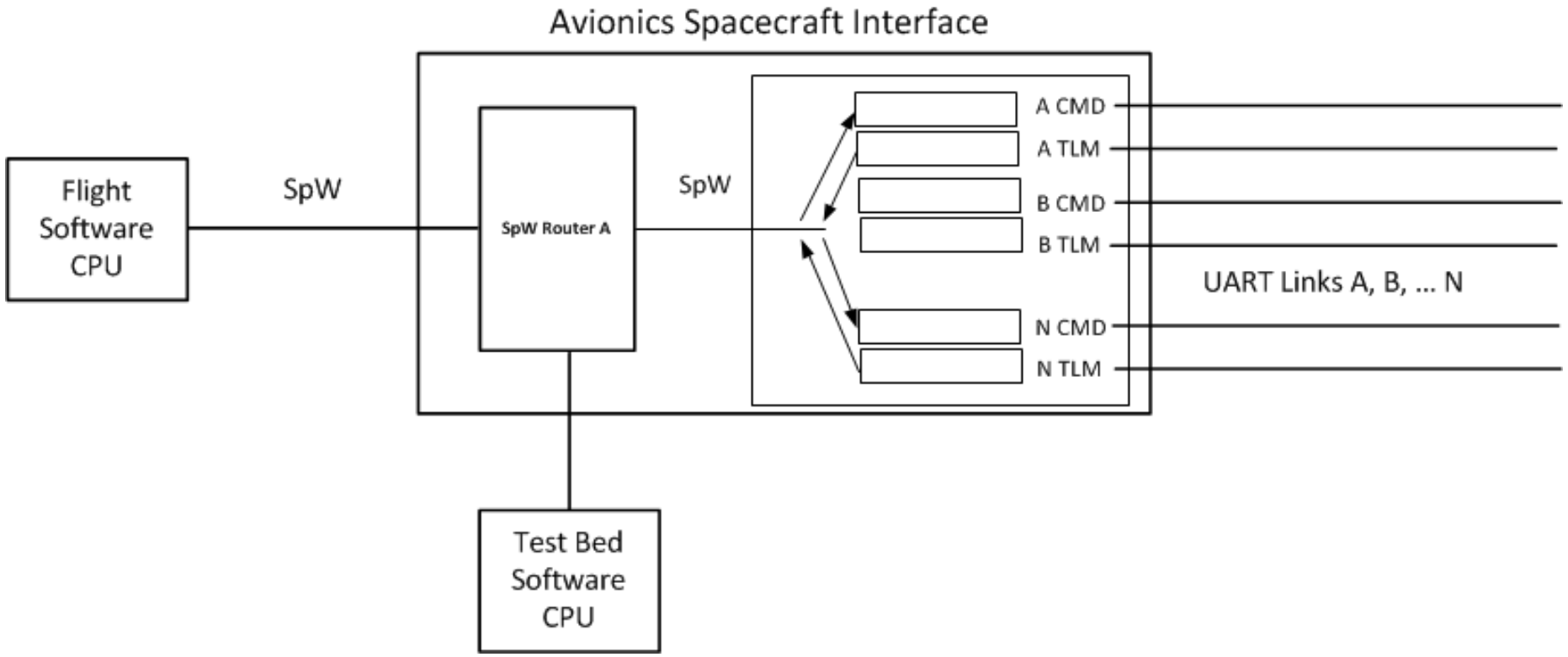
UART Links

Testbed

A CMD A In
A TLM A Out
A CMD B In
A TLM B Out
B CMD A In
B TLM A Out
B CMD B In
B TLM B Out

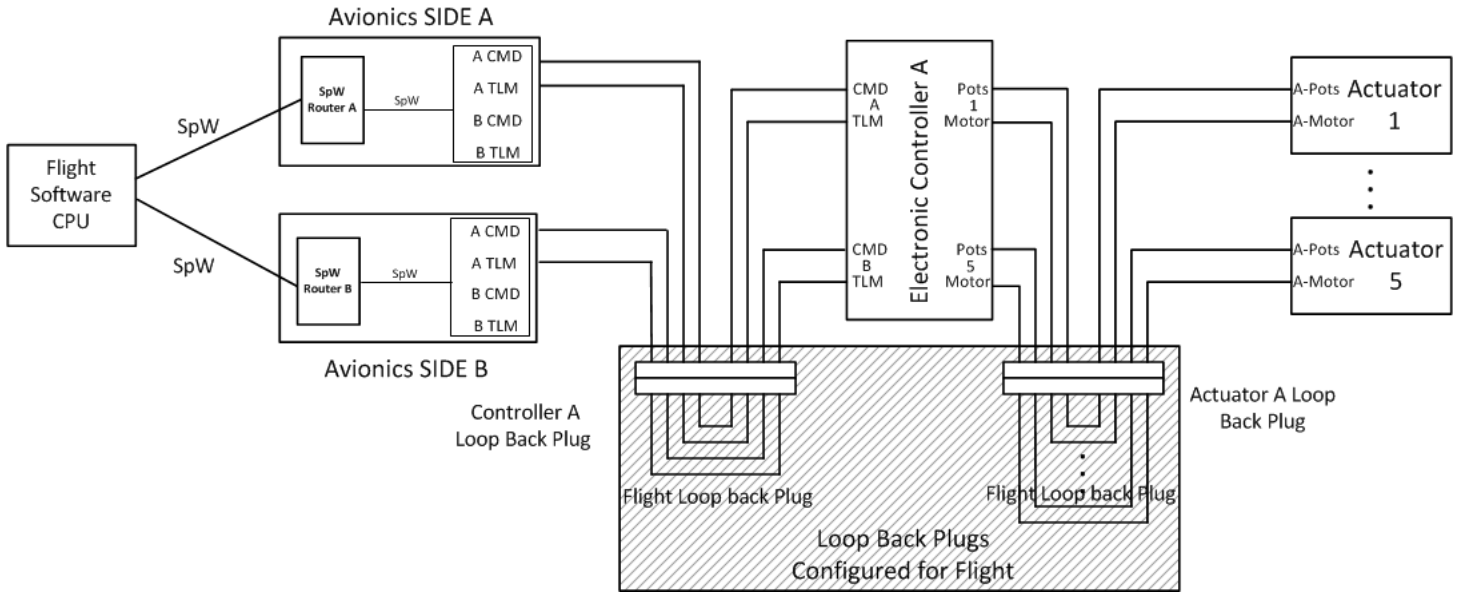
Truth Model

Spacecraft Interface Detail (Notional)



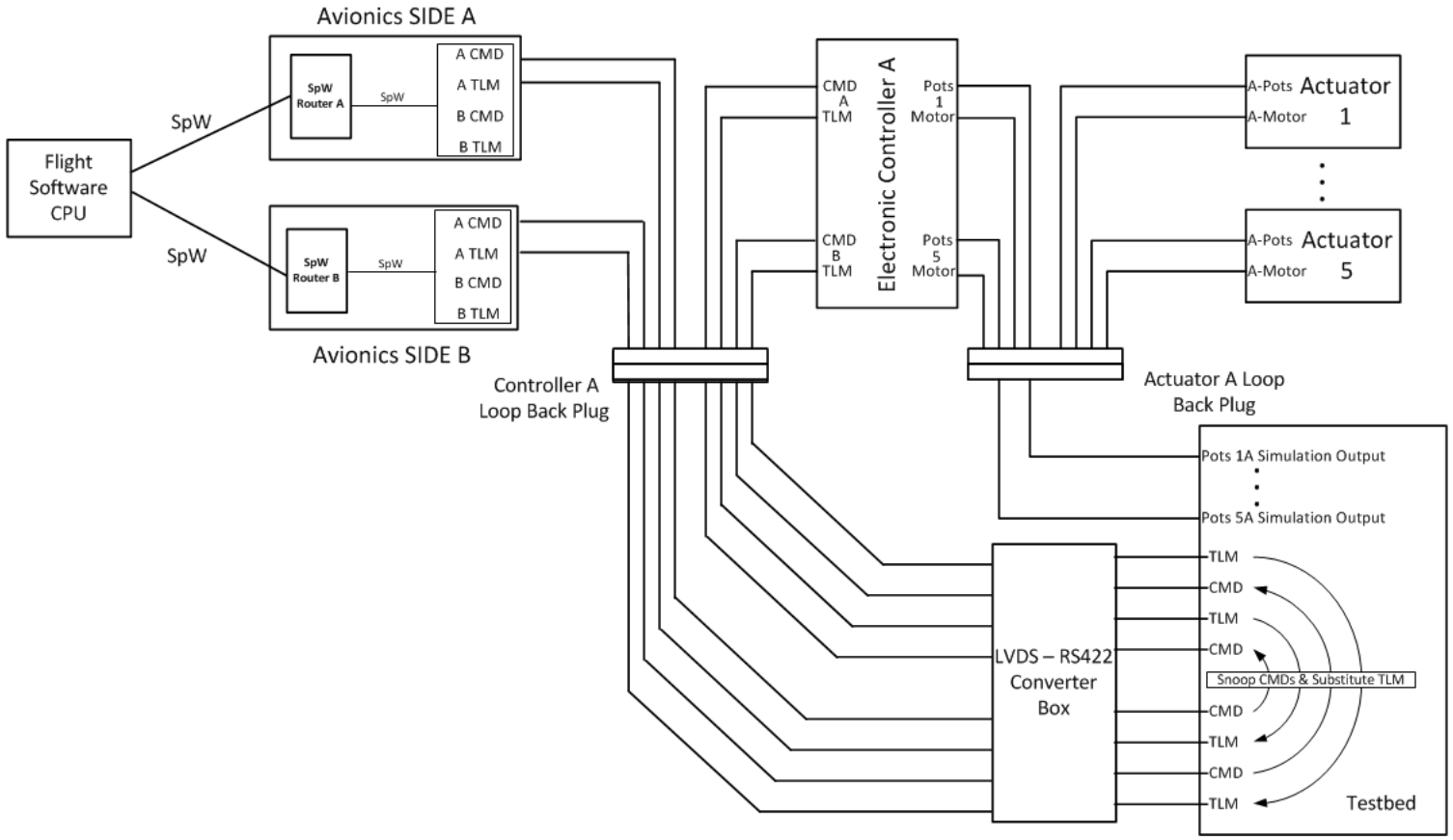
Tried and True Implementation (Notional)

Loop Back Plugs Provide Insertion Points for the Test Bed UARTS



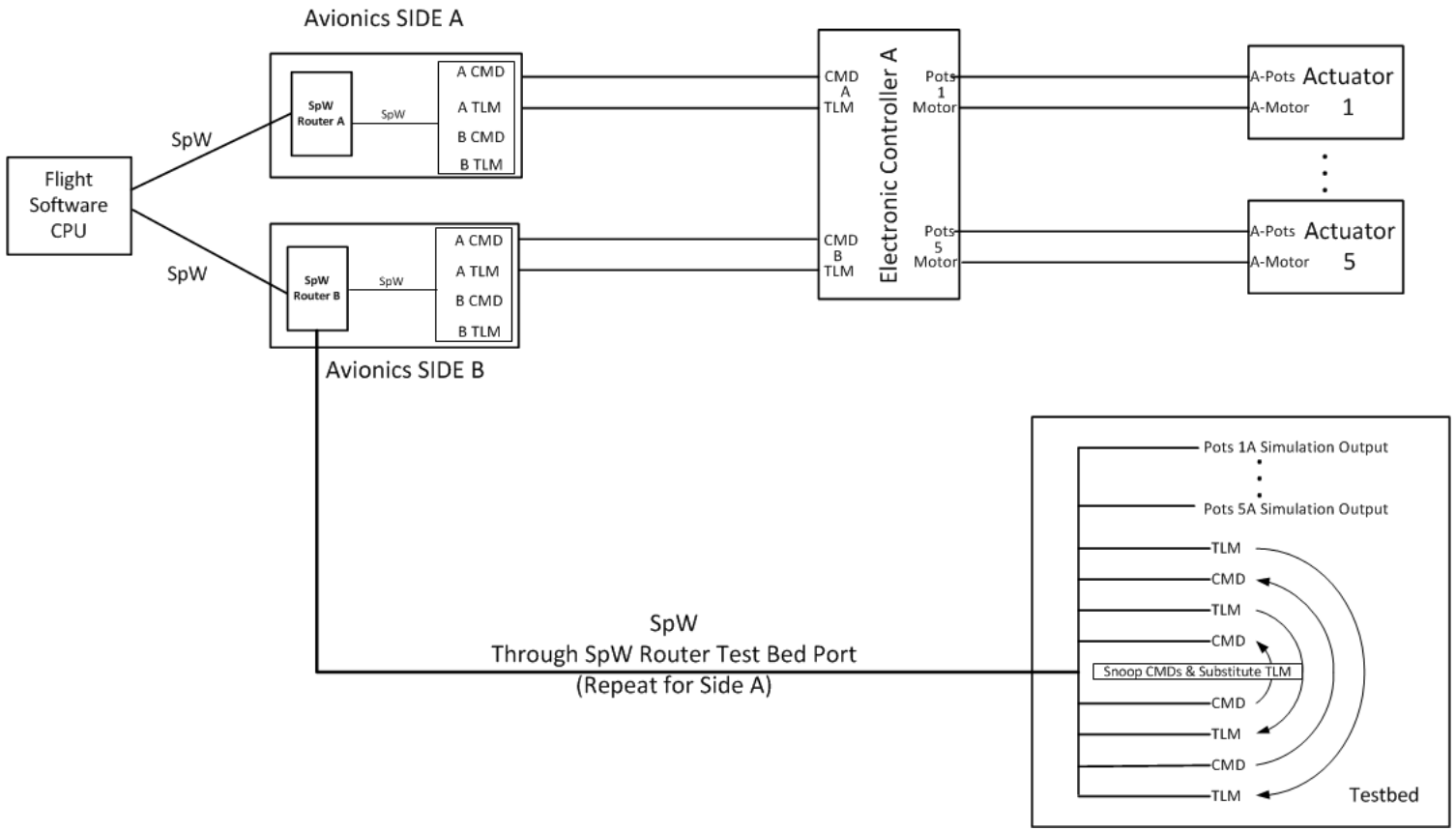
Tried and True Implementation (Notional)

Loop Back Plugs Provide Insertion Points for the Test Bed UARTS



Simplification Utilizing SpW (Notional)

Use SpW Capabilities to Snoop and Substitute through SpW Router



SpaceWire Capabilities Used

- **Remote Memory Access Protocol (RMAP)**
  - **Allows reading and writing of avionics memory buffers from multiple sources.**
    - **In flight only the flight software initiates RMAP transactions.**
    - **During system level testing the testbed software may also initiate RMAP transactions, allowing it to snoop and substitute avionics data.**
- **Logical Addressing**
  - **Each separate component, regardless of physical node, has separate logical address.**
  - **The testbed software can re-route transactions for specific components by updating the SpaceWire router logical address table for that component.**
- **Packet Duplication**
  - **Each logical address may be routed to TWO physical addresses. The packet is duplicated to the second physical address.**
  - **The testbed software can set up the router to duplicate packets from specific logical addresses allowing it to snoop avionics data in parallel with the flight software.**

Component Substitution – Utilizes Logical Addressing

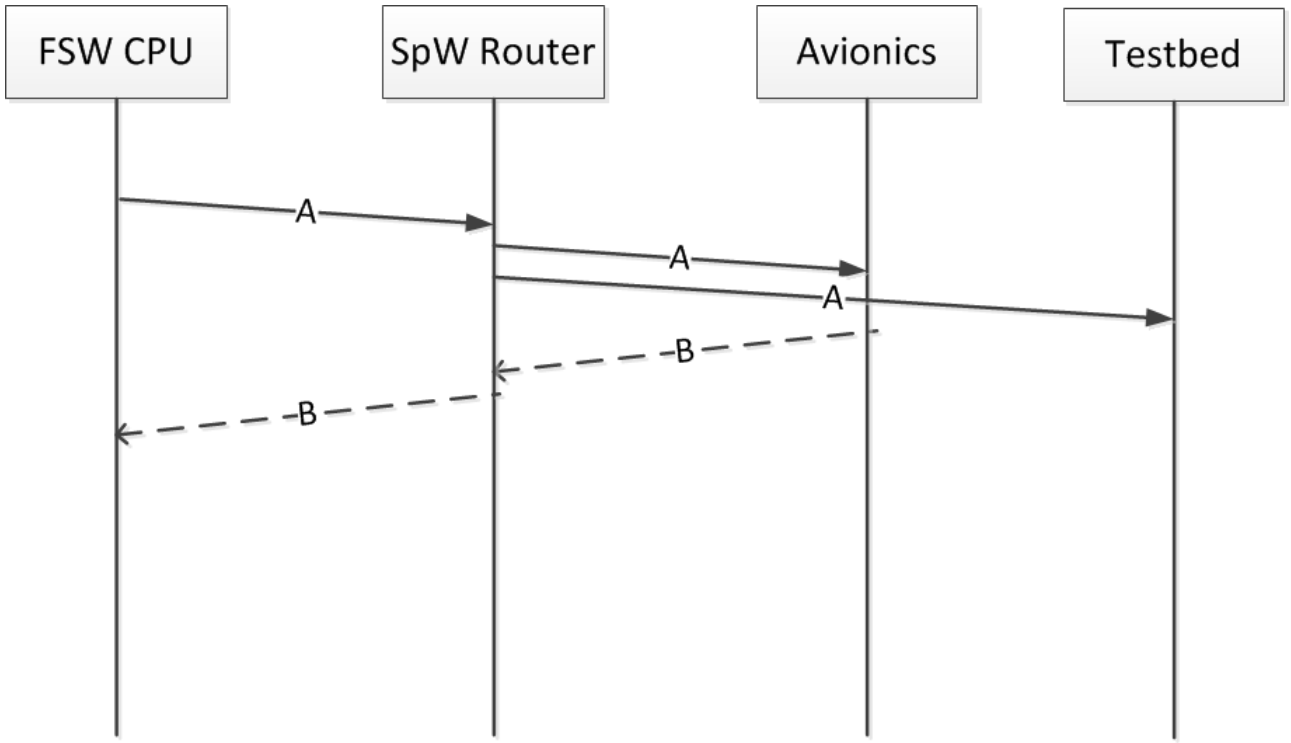
- The testbed substitutes for components that are under development or that have been “swapped out” for various reasons during integration testing.
  - The testbed software updates the SpW router to route the components logical address to the testbed, rather than the component’s port.
  - The testbed software emulates the component’s functionality and responds appropriately.
  - The flight software functions as it would in flight.

Truth Model Maintenance – Utilizes Packet Duplication

- The testbed software maintains a “truth model” that simulates the dynamics of the spacecraft based on the history of reaction wheel / thruster actuation.
  - The testbed software updates the logical addresses of the components providing sensor data in the SpaceWire routing table.
  - The update specifies the testbed SpaceWire port as a secondary destination for packets directed to that logical address.
  - The router duplicates SpaceWire packets sent to that logical address to the testbed’s SpaceWire port, allowing the testbed software to “snoop” on commands and telemetry, updating the truth model.

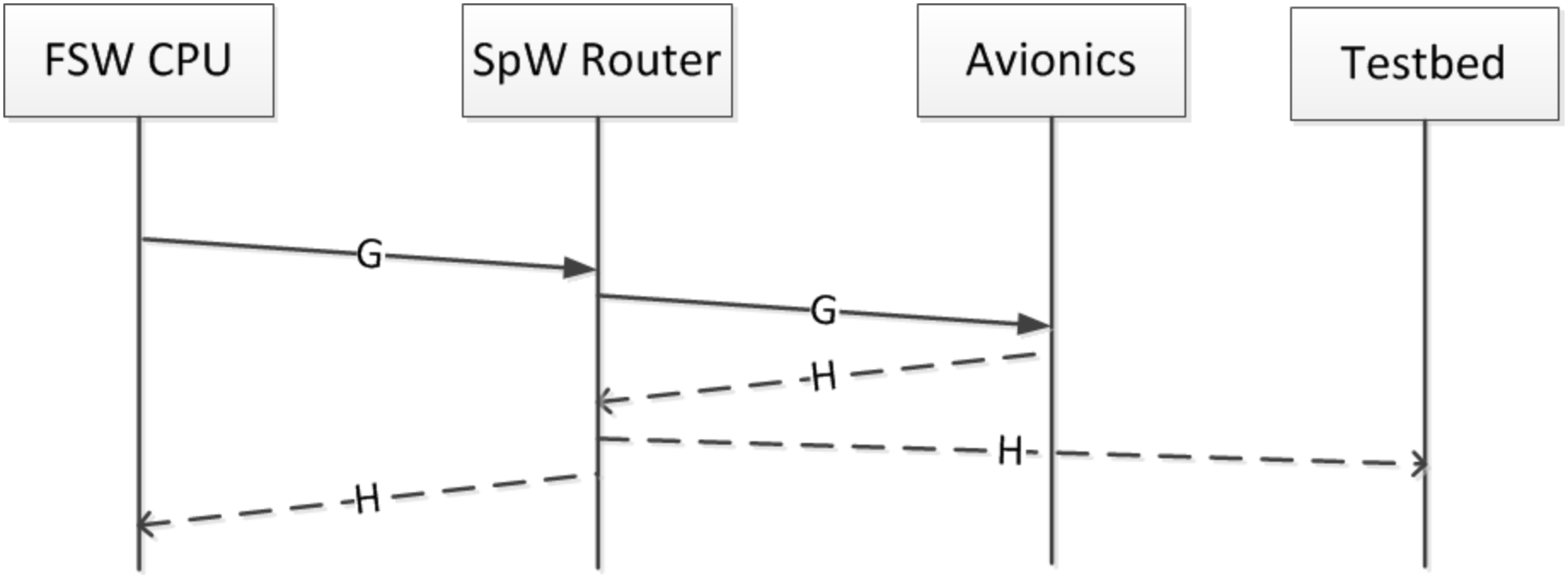
Truth Model – Command Snooping

- **Commands = RMAP Write Transactions**



Truth Model – Telemetry Snooping

- **Telemetry = RMAP Read**

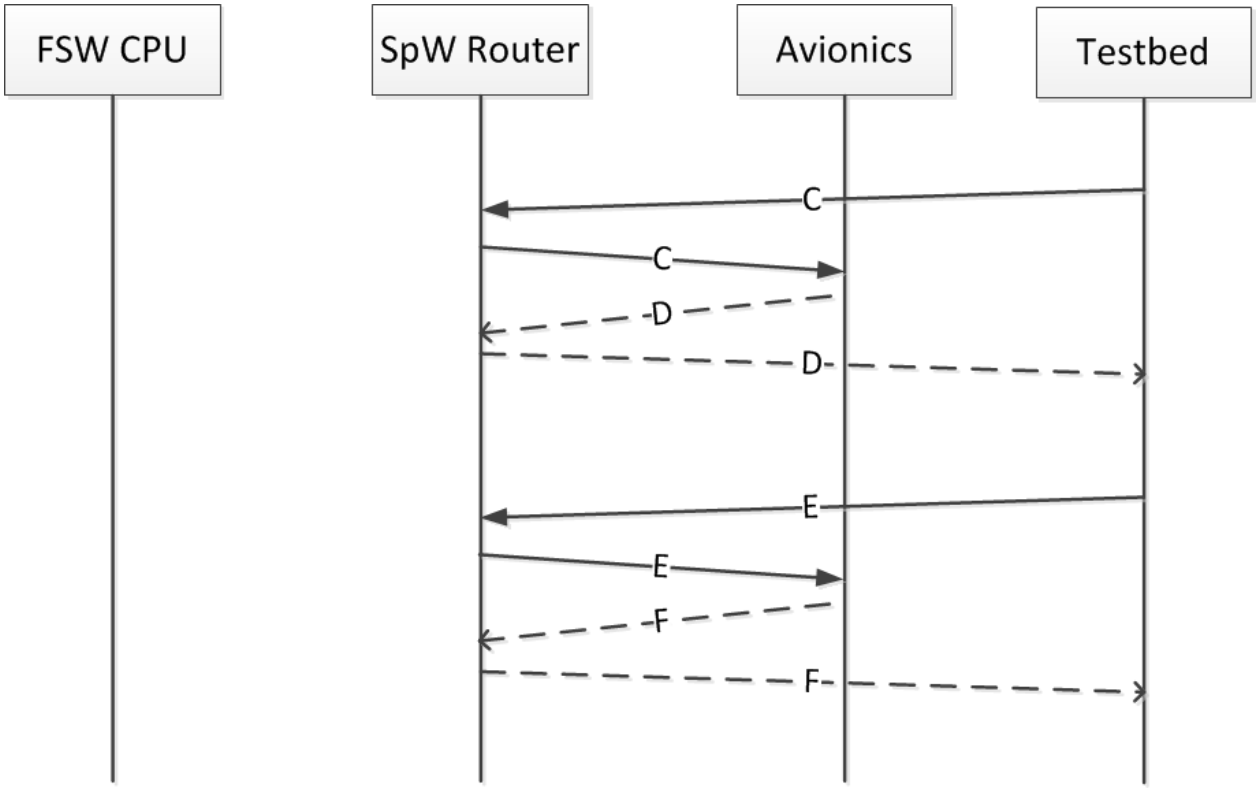


Truth / Fault Injection

- **The testbed updates avionics telemetry with the “truth” maintained by the truth model.**
- **The testbed allows injection of non-nominal data to simulate fault conditions for testing autonomy responses.**
  - **The testbed performs a “read – modify – write” sequence that substitutes the “truth” or the “fault” for the telemetry being returned by the avionics components.**
    - **RMAP Read from the avionics telemetry buffer for the component.**
    - **Testbed software uses truth model values or commanded fault value to modify the telemetry and update CRC accordingly.**
    - **RMAP Write to the avionics telemetry buffer for the component.**
    - **Flight software then reads the telemetry and reacts to the supplied dynamics or fault.**

Truth Model / Fault Injection – Telemetry Override

- **Testbed Software RMAP Read / Truth Modify / RMAP Write**



Conclusion

- **By utilizing logical addressing and packet duplication the following advantages are realized:**
  - **Allows greater fidelity to flight configuration during testing, improving "test as you fly".**
  - **Loop back connectors are not required at the UART interfaces and may be eliminated**
    - **Eliminates a mass penalty,**
    - **Simplifies design and fabrication,**
    - **Simplifies configuration changes during integration and test,**
  - **Hardware may be simulated in the test bed at the SpW level allowing temporary removals.**