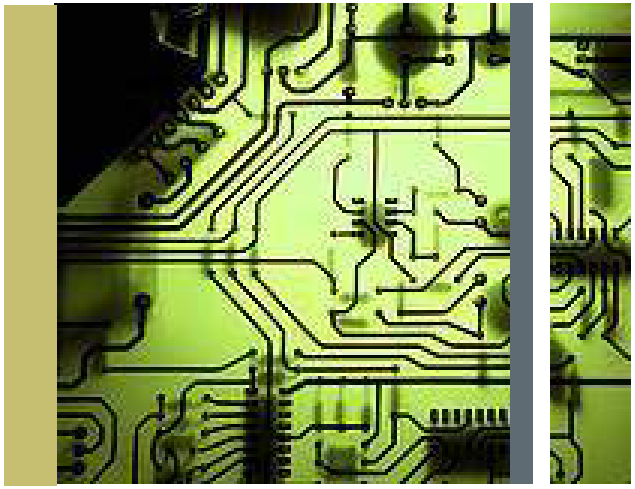
Debugging Serial Buses in Embedded System Designs

With Tektronix Oscilloscopes

Tektronix
Innovation Forum

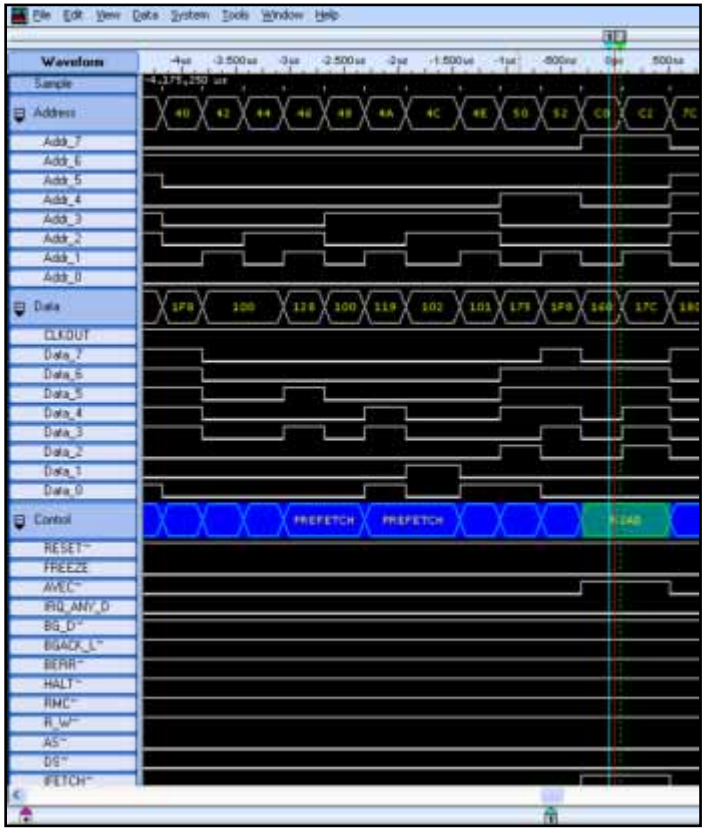
Agenda

- Introduction
- Serial Data Buses
  - I^2C
  - SPI
  - RS-232
  - CAN
  - Audio
- Troubleshooting Your Device
- Summary



Transition from Parallel to Serial Buses

- Traditional way to connect digital devices used parallel buses
- Advantages
  - Simple point-to-point connections
  - All signals are transmitted in parallel
- Disadvantages
  - Occupies a lot of circuit board space
  - All connections must be the same length
  - Many connections limit reliability
  - Connectors may be very large
- With serial buses, these disadvantages are minimized



Tektronix Innovation Forum

Design Implications of Serial Communication

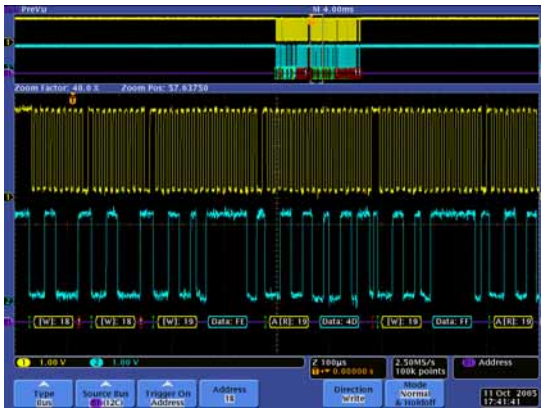
- Product Design
  - Integrated into many processors, ASICs, and FPGAs
  - Easier signal routing
  - Less space, less weight, less power
  - Higher manufacturing yields and reliability
  - ***Improves circuit board designs, lowers cost and reduces form factor***

- System Design
  - SPI and I^2C buses enable connection of many ICs without external components.
  - CAN enables connection of sub-assemblies in automotive applications, and industrial controls in factory automation applications.
  - ***Simplifies designs of complex systems***.

Debug Challenges of Serial Data

- Parallel Buses
  - Each line has its own signal path
  - Clock is generally a separate line
  - Easy to decode
  - State and Pattern triggering and decoding are straightforward with a logic analyzer or mixed signal oscilloscope

- Serial Buses
  - Signals are spread over time
  - Clock is sometimes embedded
  - Decode is tedious
  - Must decode first to trigger on packet
  - Analysis solutions available on some oscilloscopes



Serial data complicates bus troubleshooting

Tektronix Innovation Forum

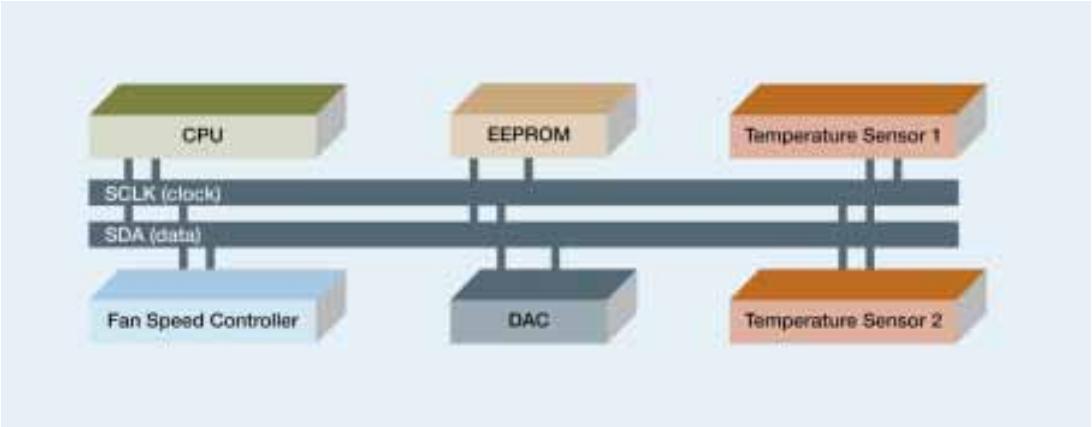
Serial Bus Review

- I^2C
- SPI
- RS-232 (RS-422, RS-485, UART)
- CAN
- Audio

Tektronix Innovation Forum

I²C (Inter-Integrated Circuit)

- Used for chip-to-chip communication between microcontrollers and A/Ds, D/As, FPGAs, sensors, etc.
- Uses two single-ended, bi-directional signals: clock and data
- Any I²C device can be attached to the bus
- Data rates:
  - Standard Mode (100 kbps)
  - Fast Mode (400 kbps)
  - High Speed Mode (3.4 Mbps)



Tektronix Innovation Forum

I²C Message Structure

- Start: Indicates the device is taking control of the bus and a message will follow
- Address: 7-bit or 10-bit number representing the device address to read or write
- Data: Integer number of bytes read from or written to the device
- Acknowledge: 1-bit from the slave device acknowledging the master's actions
- Stop: Indicating the message is complete and the master has released the bus

Tektronix Innovation Forum

SPI (System Peripheral Interface)

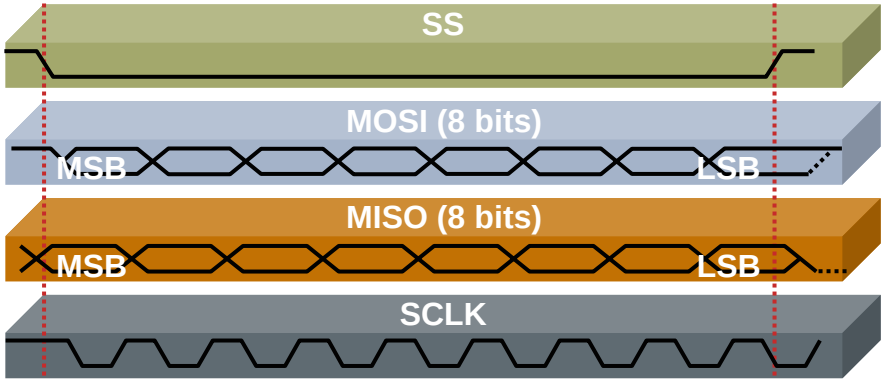
- Used primarily to communicate between microcontrollers and their immediate peripheral devices
- Typical configuration has four signals: SCLK, MOSI, MISO, SS
  - Data is simultaneously transmitted and received
  - SS line used to specify slave device
  - Each unique device on bus has its own SS signal from master
- Multiple bus configurations are allowed
  - Network can use 2-, 3-, or 4-wire bus topology
- Data rates up to 10 Mbps

SS – enables slave device to accept data

MOSI – data from the master to a slave

MISO – data from a slave to the master

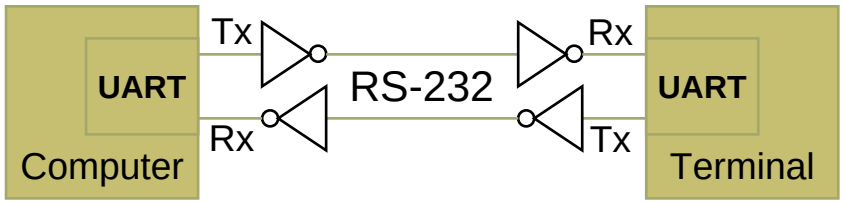
SCLK – serial clock driven by Master



Tektronix Innovation Forum

RS-232 (Recommended Standard-232)

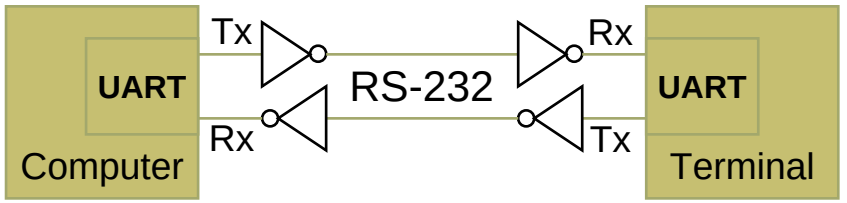
- Point-to-point communication at slow speeds over short distances
- Two single-ended signals provide point-to-point, full-duplex communication
- Standard does not specify character encoding, data framing, or protocols
- Transmission systems:
  - Managed by Universal Asynchronous Receiver/Transmitters (UARTs)
    - Pre-determined bit rate
  - RS-232 is an inverting, single-ended high-voltage interface



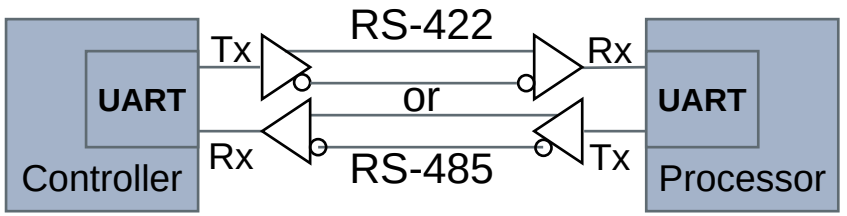
RS-232 Application Example

RS-232 (Recommended Standard-232)

- Point-to-point communication at slow speeds over short distances
- Two single-ended signals provide point-to-point, full-duplex communication
- Standard does not specify character encoding, data framing, or protocols
- Transmission systems:
  - Managed by Universal Asynchronous Receiver/Transmitters (UARTs)
    - Pre-determined bit rate
  - RS-232 is an inverting, single-ended high-voltage interface
  - RS-422 or RS-485 are differential interfaces



RS-232 Application Example

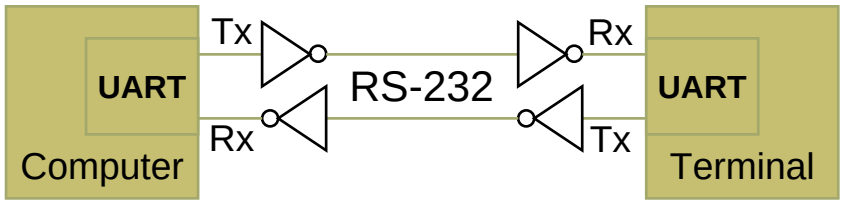


Audio and Video Application Example

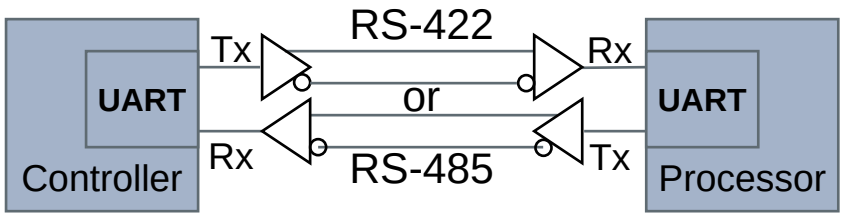
Tektronix Innovation Forum

RS-232 (Recommended Standard-232)

- Point-to-point communication at slow speeds over short distances
- Two single-ended signals provide point-to-point, full-duplex communication
- Standard does not specify character encoding, data framing, or protocols
- Transmission systems:
  - Managed by Universal Asynchronous Receiver/Transmitters (UARTs)
    - Pre-determined bit rate
  - RS-232 is an inverting, single-ended high-voltage interface
  - RS-422 or RS-485 are differential interfaces
  - Or ICs can be connected directly



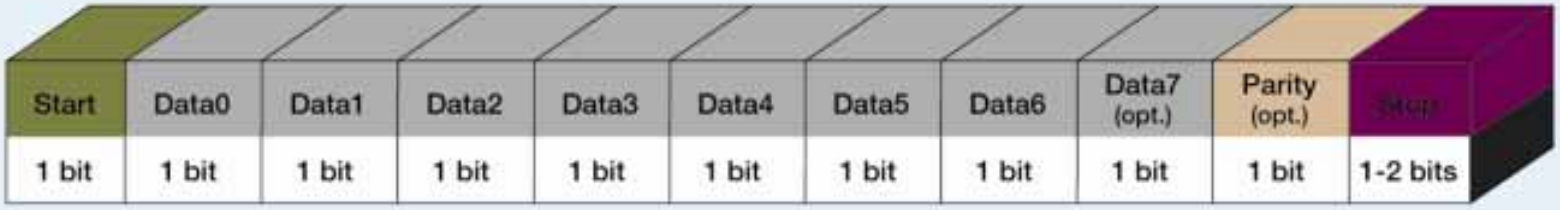
RS-232 Application Example



Audio and Video Application Example

Embedded Communication Application Example

RS-232 Byte Structure

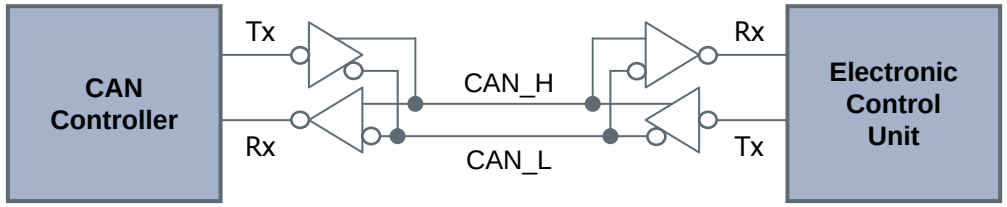


- Each character begins with a Start bit, a logic “0”
- Character is comprised of 7 or 8 data bits
- Optional Parity bit is next
- Terminated in 1, 1.5, or 2 stop bits

Tektronix Innovation Forum

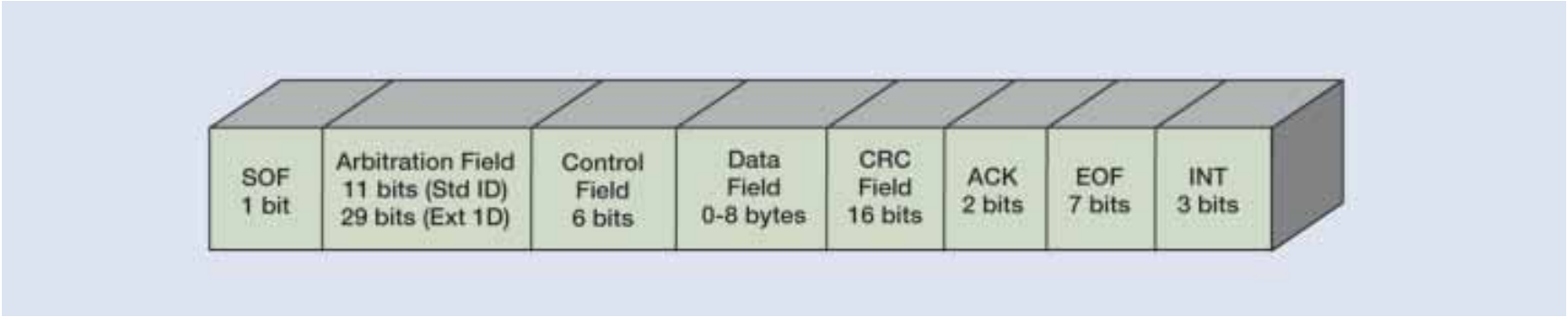
CAN (Controller Area Network)

- Used for system-to-system communication in Automotive, Industrial Automation, and Medical Equipment
- Serial asynchronous, multi-master, layered communication network
  - Sophisticated error detection and error handling mechanisms
  - Flexible signaling support for low-cost implementation
  - Messages are broadcast to all nodes on the network
- Physical bus is single-wire or dual-wire, and fault tolerant
- Data rates from 5 kbps to 1 Mbps



CAN Physical Layer

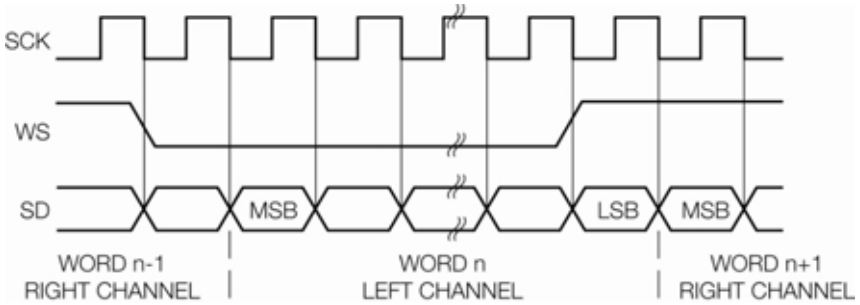
CAN Data and Remote Frame Overview



- SOF: begins with a start of frame (SOF) bit
- Arbitration: Identifier (address) and Remote Transmission Request (RTR) bit
- Control: 6 bits including Identifier Extension (IDE) bit and Data Length Code (DLC)
- Data: zero to eight bytes of data
- CRC: 15-bit cyclic redundancy check code and a recessive delimiter bit
- ACK: acknowledge field is two bits long
- EOF: 7 recessive bits indicate the end of frame (EOF)
- INT: intermission field of three recessive bits indicates the bus is free

I²S (Inter-IC Sound) and Derivatives

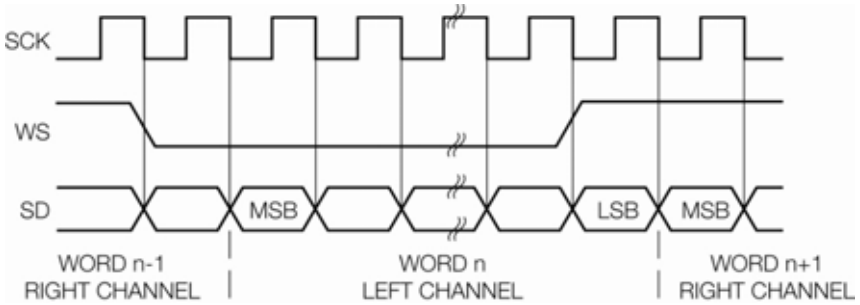
- Used for communicating serial audio data between ICs
  - Inter-IC Sound (I²S)
  - Left Justified (LJ)
  - Right Justified (RJ)
  - Time Division Multiplexed (TDM)
- Synchronous, bi-directional serial interfaces
  - High-resolution, low cost, low jitter data transmission
  - Primary differences are in timing
- Physical bus consists of three lines: Bit Clock, Word Select, and Data
- Data rates up to a few Megabits/second



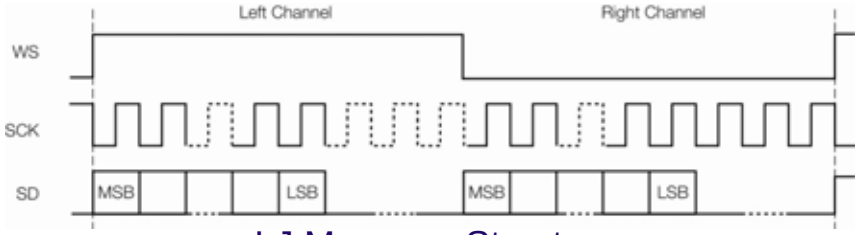
I²S Message Structure

I²S (Inter-IC Sound) and Derivatives

- Used for communicating serial audio data between ICs
  - Inter-IC Sound (I²S)
  - Left Justified (LJ)
  - Right Justified (RJ)
  - Time Division Multiplexed (TDM)
- Synchronous, bi-directional serial interfaces
  - High-resolution, low cost, low jitter data transmission
  - Primary differences are in timing
- Physical bus consists of three lines: Bit Clock, Word Select, and Data
- Data rates up to a few Megabits/second



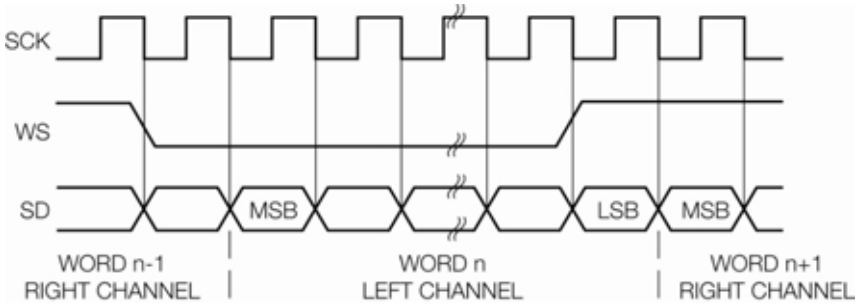
I²S Message Structure



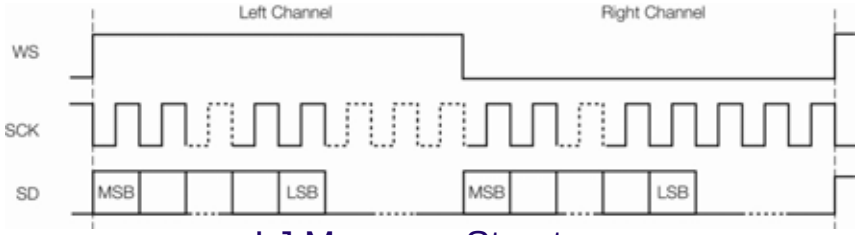
LJ Message Structure

I²S (Inter-IC Sound) and Derivatives

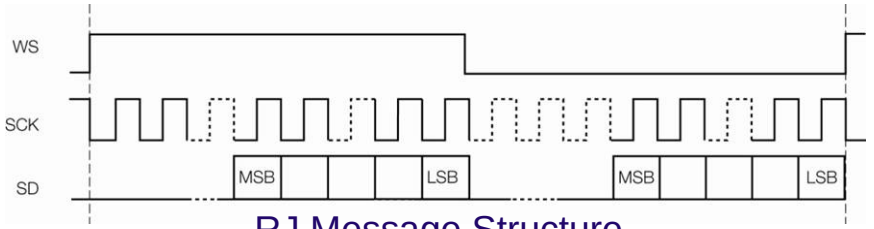
- Used for communicating serial audio data between ICs
  - Inter-IC Sound (I²S)
  - Left Justified (LJ)
  - Right Justified (RJ)
  - Time Division Multiplexed (TDM)
- Synchronous, bi-directional serial interfaces
  - High-resolution, low cost, low jitter data transmission
  - Primary differences are in timing
- Physical bus consists of three lines: Bit Clock, Word Select, and Data
- Data rates up to a few Megabits/second



I²S Message Structure



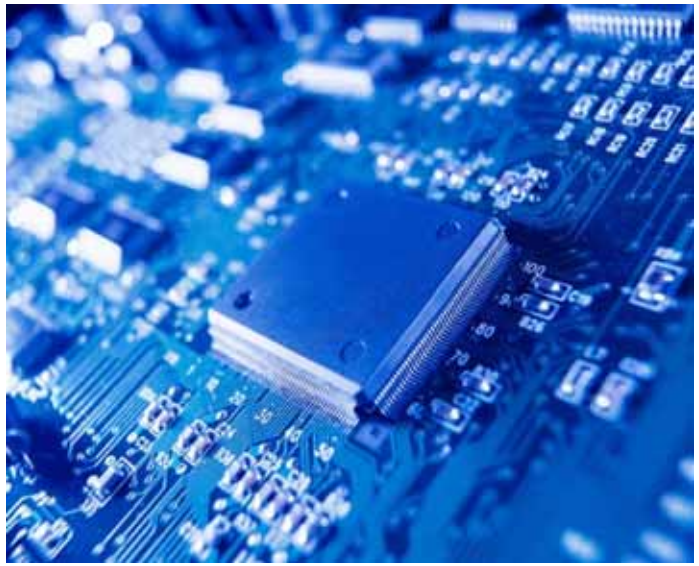
LJ Message Structure



RJ Message Structure

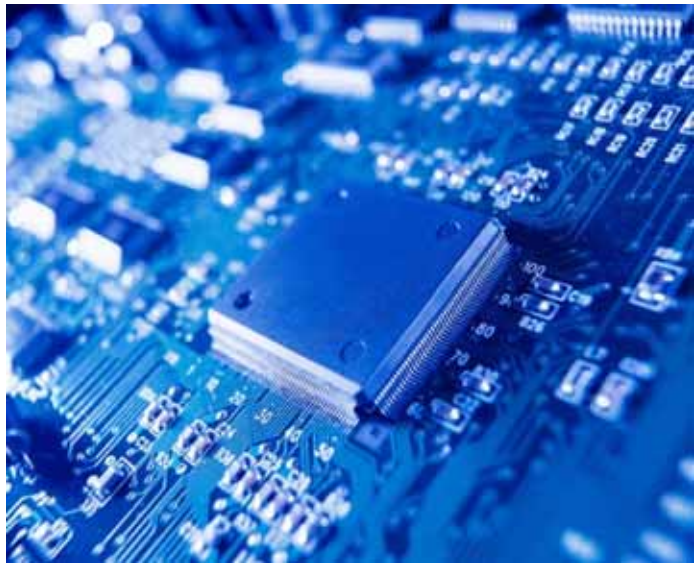
Troubleshooting Your Device

- Decode of Serial Data
- Capture and Search for Specific Messages
- Characterize System Timing
- Trace Data Flow Through a Network
- In-Depth Analysis of Network Performance



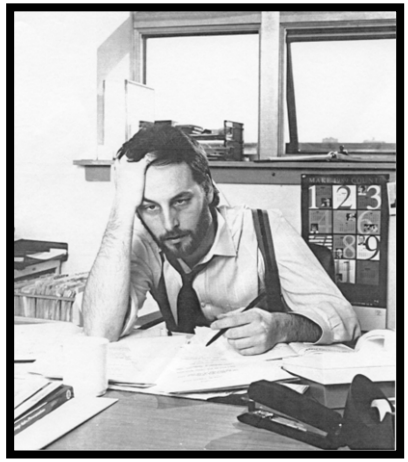
Troubleshooting Your Device

- Decode of Serial Data
- Capture and Search for Specific Messages
- Characterize System Timing
- Trace Data Flow Through a Network
- In-Depth Analysis of Network Performance



Decode of Serial Data

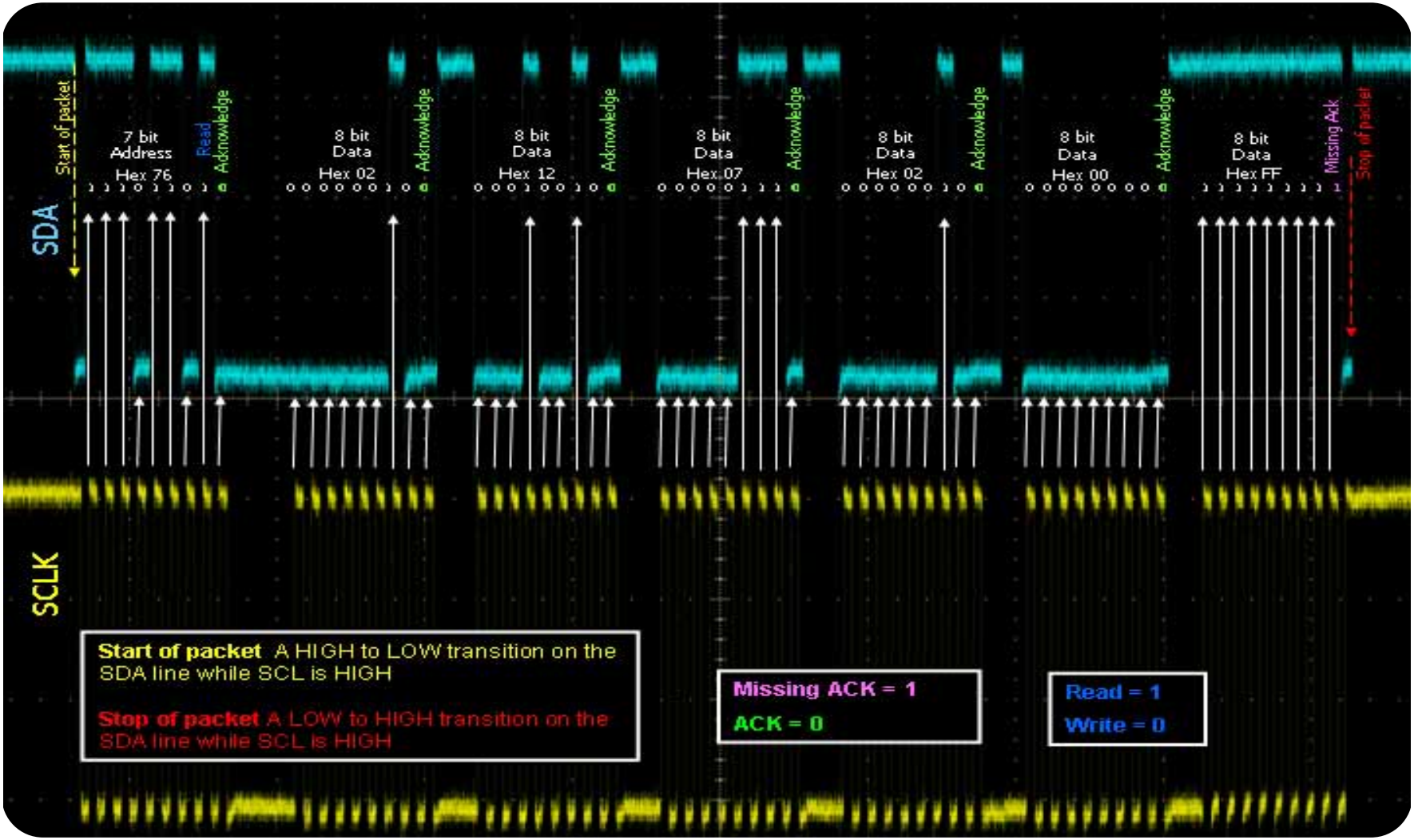
- ***Hardware Engineers:*** verify connections and adequate signal integrity for the bus to transmit data.
  - Monitor waveforms and decoded bus data values
- ***Software/Firmware Engineers:*** verify bus messages are being sent as expected.
  - Waveform displays are not the preferred format.
- ***System Engineers:*** verify system components are working together as designed.
  - Again, waveform displays are not the preferred format.



Bus waveforms can be manually decoded…

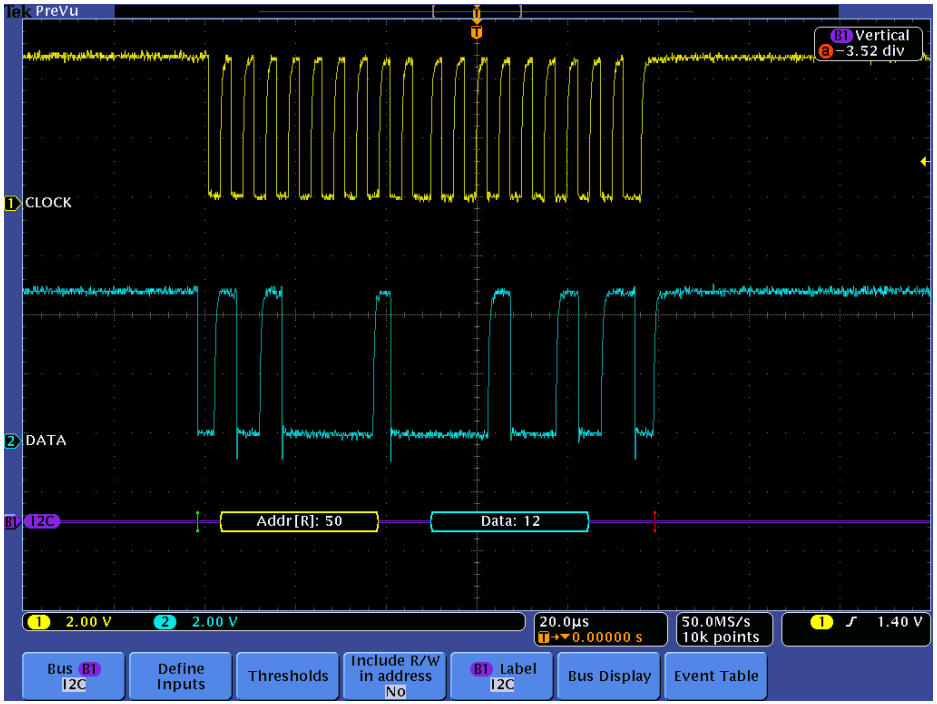
But it is tedious and error-prone.

Example - I²C Manual Decoding

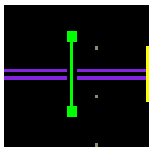


Innovation Forum

Automated Decode with Tektronix' Oscilloscopes



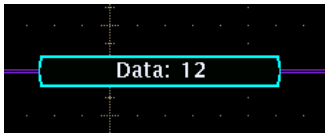
Tektronix MSO/DPO4000 Series Automated Decode



Start

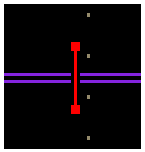
Address

[W] for Write, [R] for Read
Displayed in hex or binary



Data

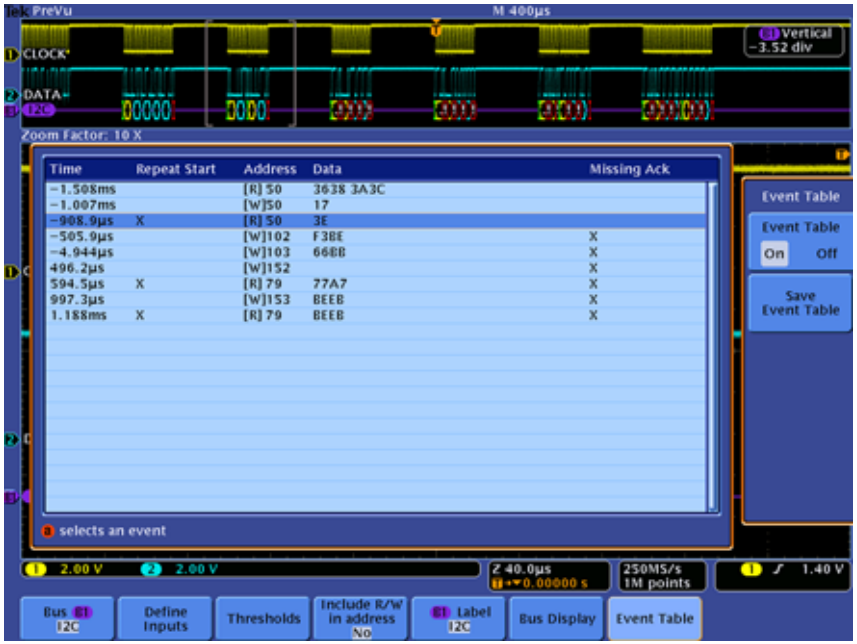
Displayed in hex or binary



Stop

Event Table for Viewing Bus Traffic

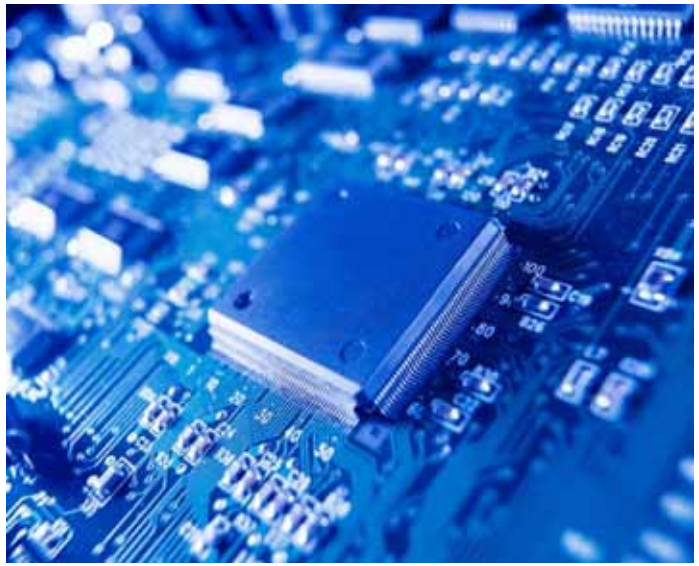
- Shows decoded message content with time stamps
- View bus traffic in tabular format
- Compare with software listings
- Easy timing measurements



Tektronix MSO/DPO4000 Series Event Table

Troubleshooting Your Device

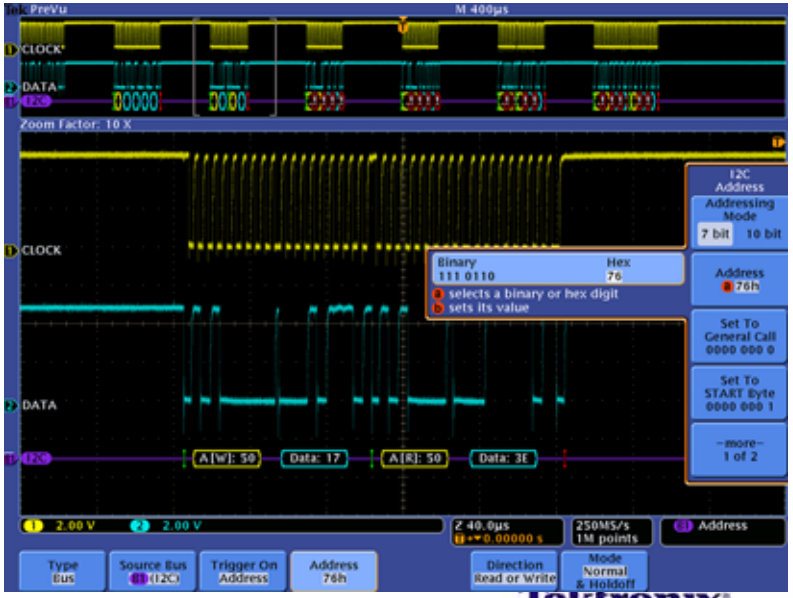
- Decode of Serial Data
- Capture and Search for Specific Messages
- Characterize System Timing
- Trace Data Flow Through a Network
- In-Depth Analysis of Network Performance



Capture a Specific Message

- Even if you can easily decode messages, the message of interest probably wasn't captured
- Need to specify messages to ***capture***:
  - In the language of the serial bus standard
  - On all critical elements of the serial message
  - With full or partial specification
- Tektronix' oscilloscopes offer serial data triggers

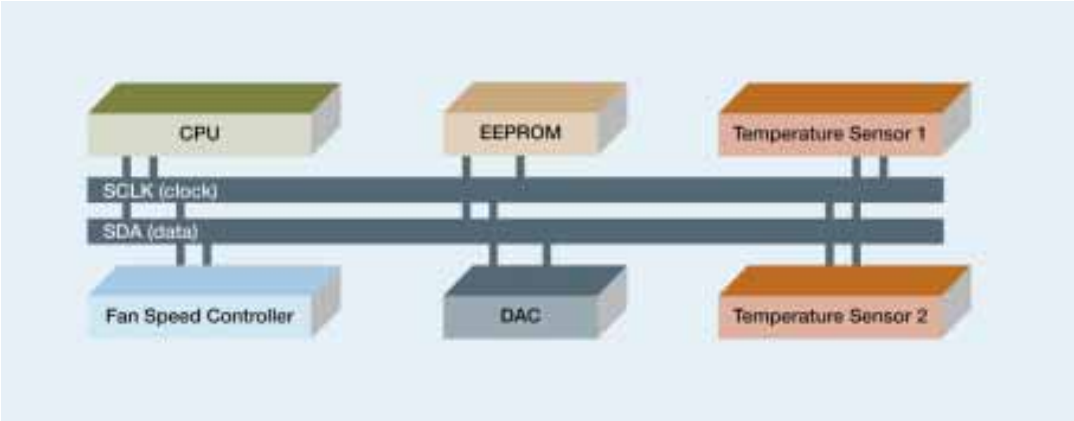
Tektronix
MSO/DPO4000 Series
I²C Triggers



Tektronix Innovation Forum

An Example: Faulty Thermal Management System

- The product is overheating and shutting off.
- Microprocessor-controlled thermal management system should sense the product's internal temperature and adjust the fan speed.
  - All of the circuits appear to have the correct power applied.
  - The processor is running and appears to be communicating with the sensors and the fan control module.
  - The software team is sure that the software is running as designed.
- Yet, the product is getting hot and the fan is not turning on.



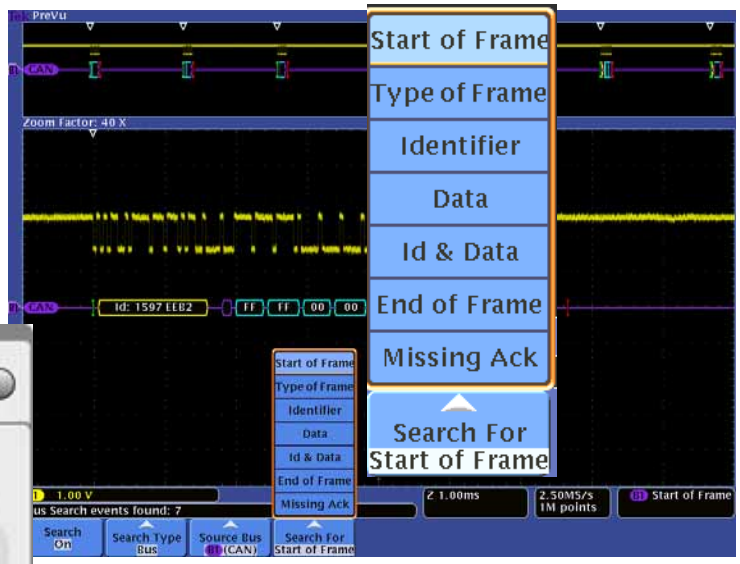
Trigger on Packet Content

- Trigger on address 18 (sensor).
- Software tries to communicate with the sensor – twice!
- No response.
- Moves to the next address, as designed.
- Upon close inspection of the board, a cold solder joint was found on the fan controller IC.

Tektronix Innovation Forum

Search for a Specific Message

- Even if you capture the message of interest, now you have to find it!
- Need to specify messages to ***search*** for:
  - In the language of the serial bus standard
  - On all critical elements of the serial message
  - With full or partial specification
  - Same conditions as needed for ***capture***
- Tektronix' oscilloscopes provide automatic search and mark capabilities

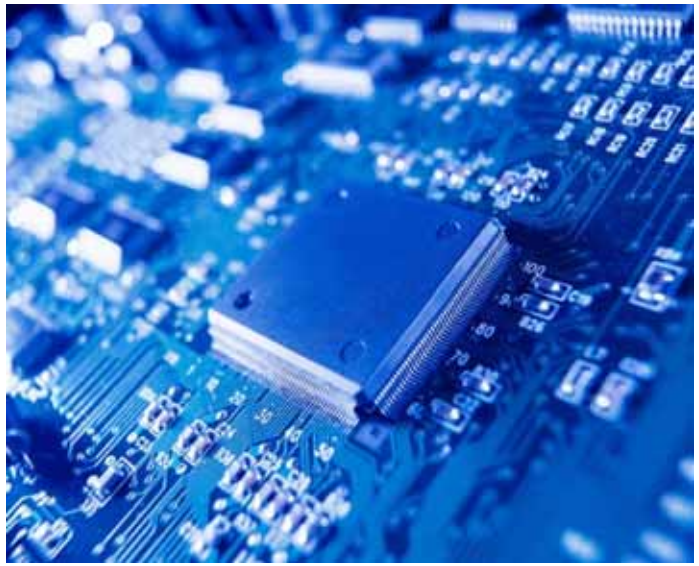


Tektronix
MSO/DPO4000 Series
Automated Search

Tektronix Innovation Forum

Troubleshooting Your Device

- Decode of Serial Data
- Capture and Search for Specific Messages
- Characterize System Timing
- Trace Data Flow Through a Network
- In-Depth Analysis of Network Performance

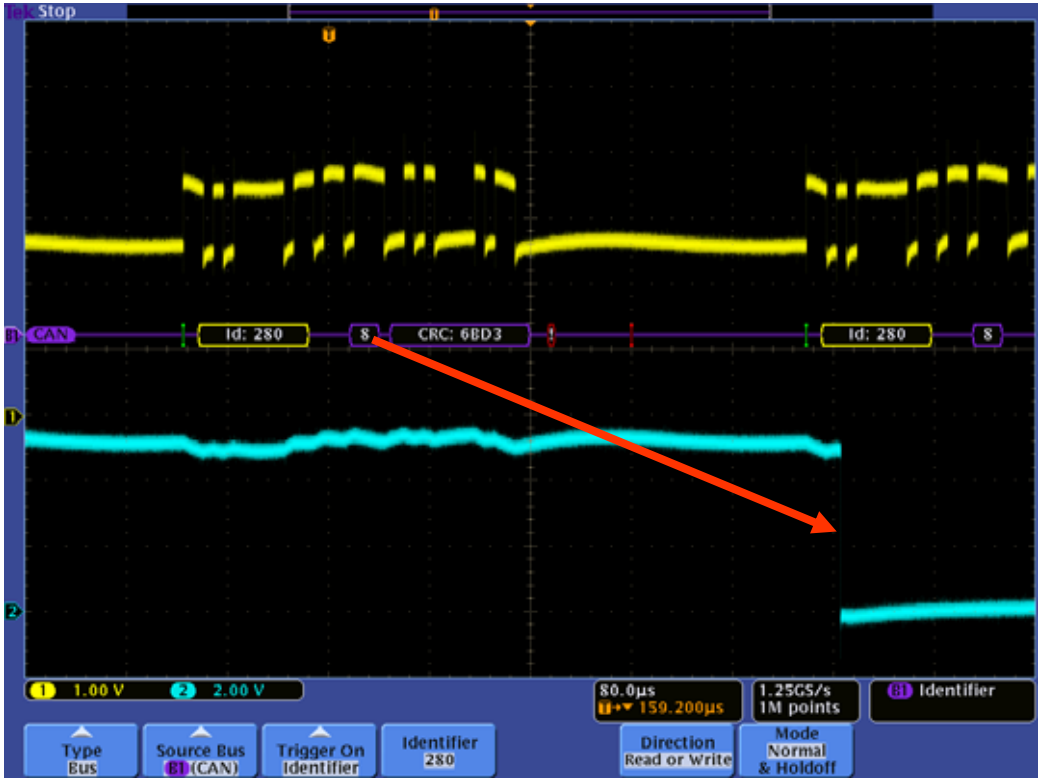


Characterize System Timing

- Characterize timing between bus messages and system operation
  - Requires waveform displays time-correlated with decoded messages
- Characterize timing differences which occur when adding a new network node to an existing network
- Automotive application example:
  - Measure worst-case time from crash sensor output to airbag activation
  - Measure variations in timing of airbag activation with varying levels of CAN bus traffic

Characterizing System Timing with Tektronix' Oscilloscopes

- Tektronix' oscilloscopes provide integrated tools for characterizing timing between bus messages and system operation:
  - Time-correlated waveform displays and decoded bus messages
  - Intensity-graded infinite persistence displays to show variations in timing

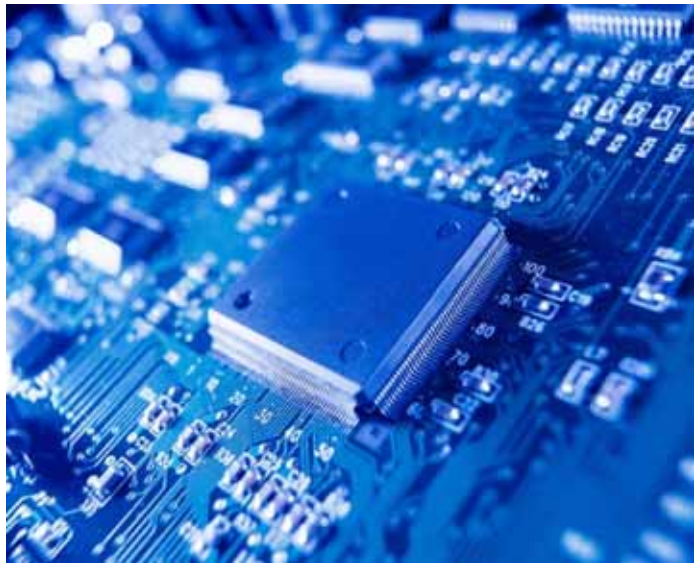


Tektronix MSO/DPO4000 Series

Tektronix Innovation Forum

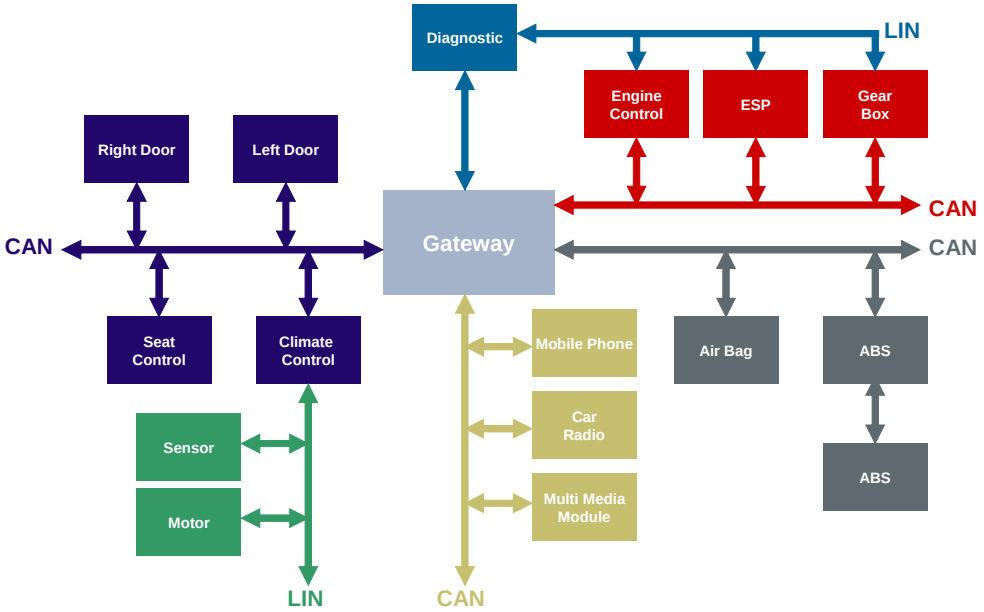
Troubleshooting Your Device

- Decode of Serial Data
- Capture and Search for Specific Messages
- Characterize System Timing
- Trace Data Flow Through a Network
- In-Depth Analysis of Network Performance



Trace Data Flow Through a Network

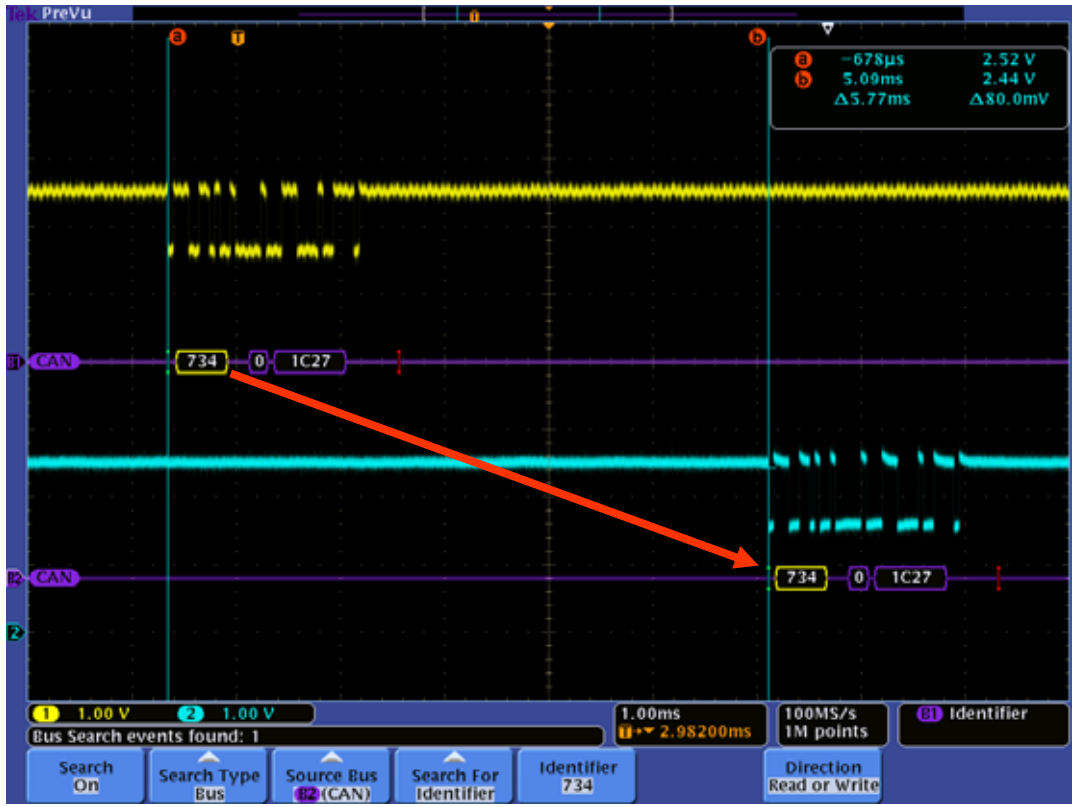
- Trace serial data flow between nodes through a network
  - Simultaneously display messages at transmitter and receiver to verify continuity and propagation delays
- Trace serial data flow between network segments separated by a gateway
  - Simultaneously display messages from multiple buses, at different speeds, or even different bus standards



Tektronix Innovation Forum

Tracing Data Flow with Tektronix' Oscilloscopes

- Tektronix' oscilloscopes simultaneously display messages at different points in the network
  - Verify continuity and propagation delays on up to 4 buses
  - Validate network gateway operation by decoding different bus speeds & protocols

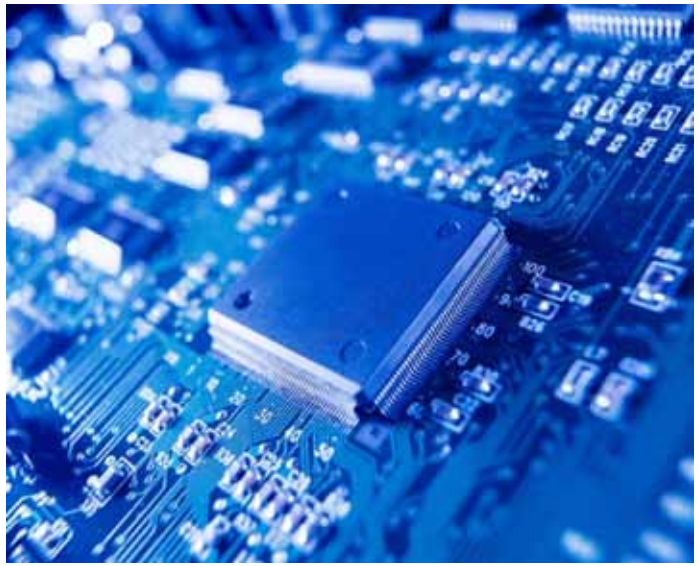


Tektronix MSO/DPO4000 Series

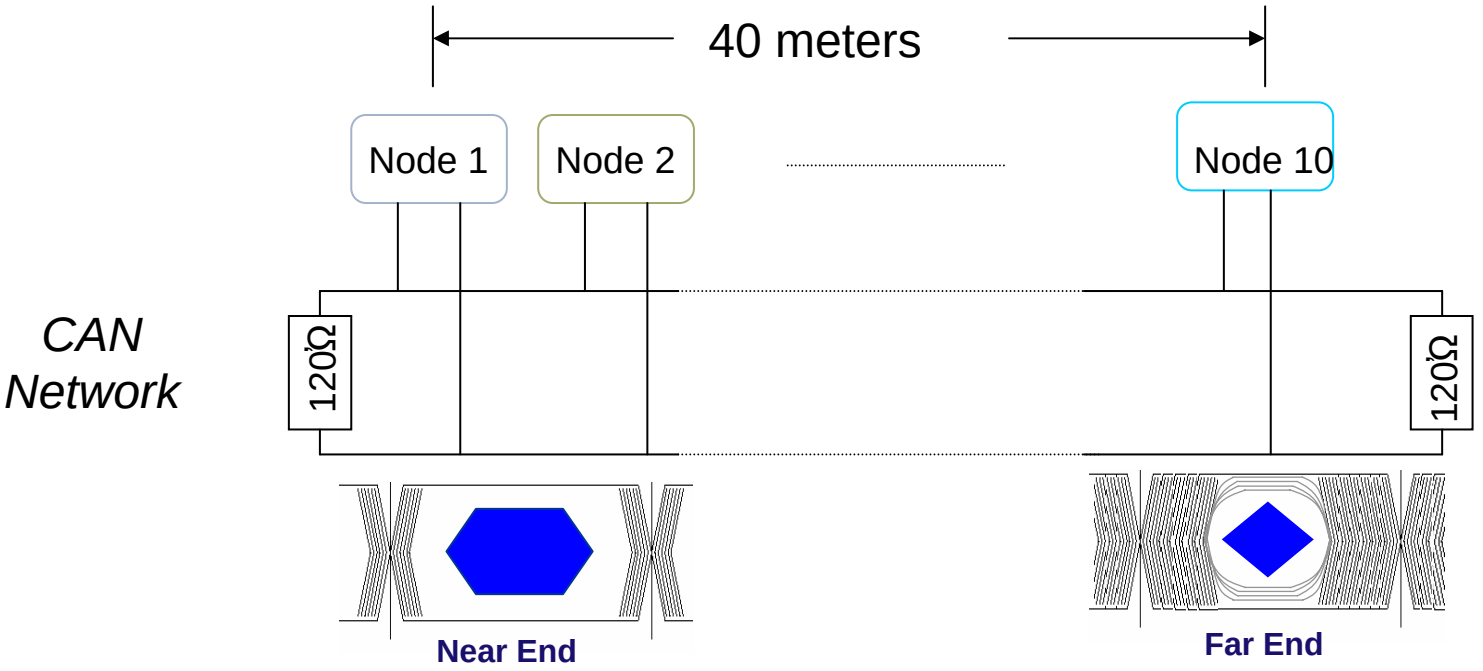
Tektronix Innovation Forum

Troubleshooting Your Device

- Decode of Serial Data
- Capture and Search for Specific Messages
- Characterize System Timing
- Trace Data Flow Through a Network
- In-Depth Analysis of Network Performance



In-Depth Analysis of Network Performance



- Locate and analyze signal integrity problems with eye diagrams
- Characterize different oscillator tolerances and propagation delays between nodes for synchronizing the network
- Monitor bus utilization to ensure efficient use of the network

Eye Diagram Analysis with Tektronix' Oscilloscopes

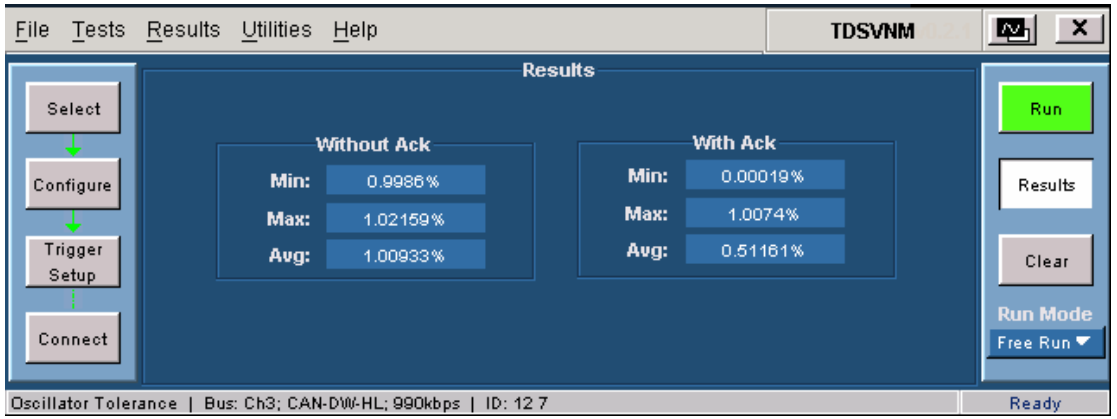
- Quickly locate noise caused by jitter, amplitude aberrations, spikes and glitches
  - Eye diagram shows changes in amplitude and jitter in the CAN bus signal
  - Measure amplitude and jitter with cursors

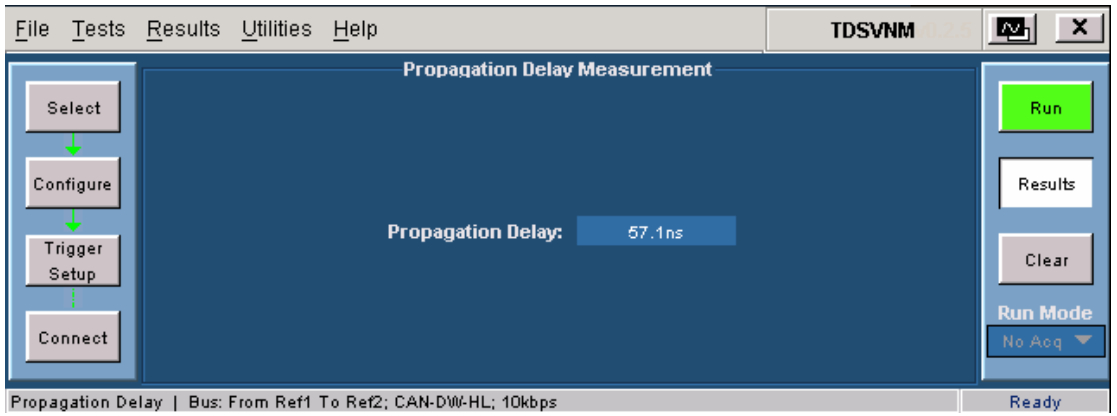
Tektronix DPO7000 Series

Tektronix Innovation Forum

Characterizing Oscillator Tolerance and Propagation Delay with a Tektronix' Oscilloscope

- Oscillator tolerance of a CAN node
  - Specify the specific ID for trigger condition
  - Result will include ACK and without ACK bit
  - With ACK bit, shows the impact of receiving CAN node oscillator tolerance on transmitting node
- Propagation Delay
  - Connect two channels to any two CAN nodes
  - Result is directly available

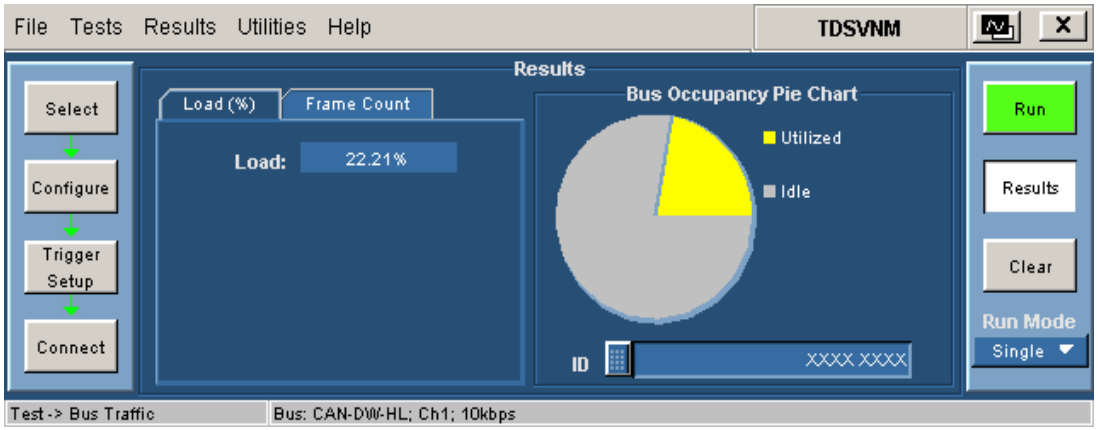


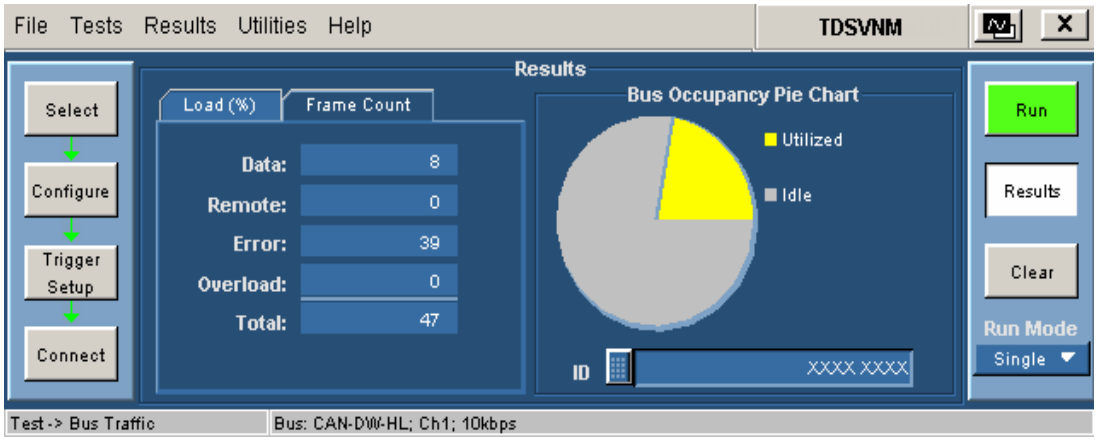


Tektronix DPO7000 Series with TDSVNM option

Monitoring CAN Traffic for Bus Utilization with a Tektronix' Oscilloscope

- Measure at specific ID, error frame or overload frame
- Specifies percentage of time traffic present in the CAN bus
- Type of traffic can be analyzed
  - Frame count



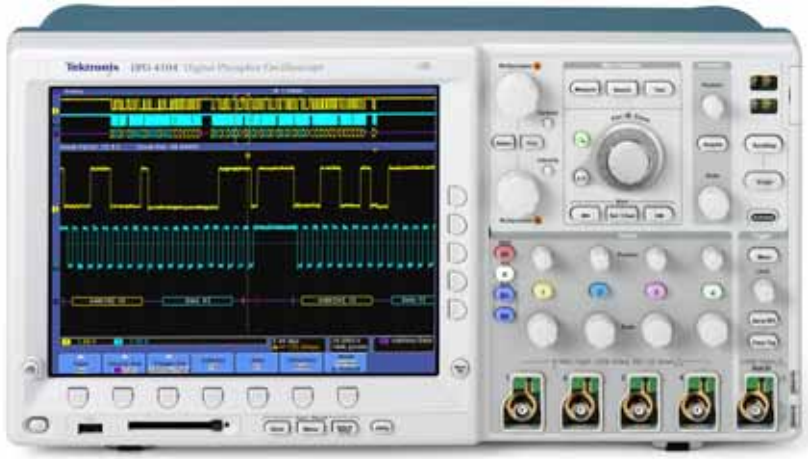


Tektronix DPO7000 Series with TDSVNM option

Summary

- Serial buses are pervasive, creating a unique set of measurement and analysis needs
- Making measurements needs to be easier, faster, and more accurate
- Requires an oscilloscope with triggering, decoding, and analysis tools for serial protocols

Tektronix' oscilloscopes offer automated decode, trigger, search and analysis for pervasive serial buses



Debugging Serial Buses with the MSO/DPO Series

Automated Decode, Trigger and Search

MSO/DPO2000 Series

- 100 MHz and 200 MHz
- 1 M record length
- Wave Inspector
- Supported serial buses:
  - I^2C
  - SPI
  - CAN
  - LIN
  - RS-232/422/485

DPO3000 Series

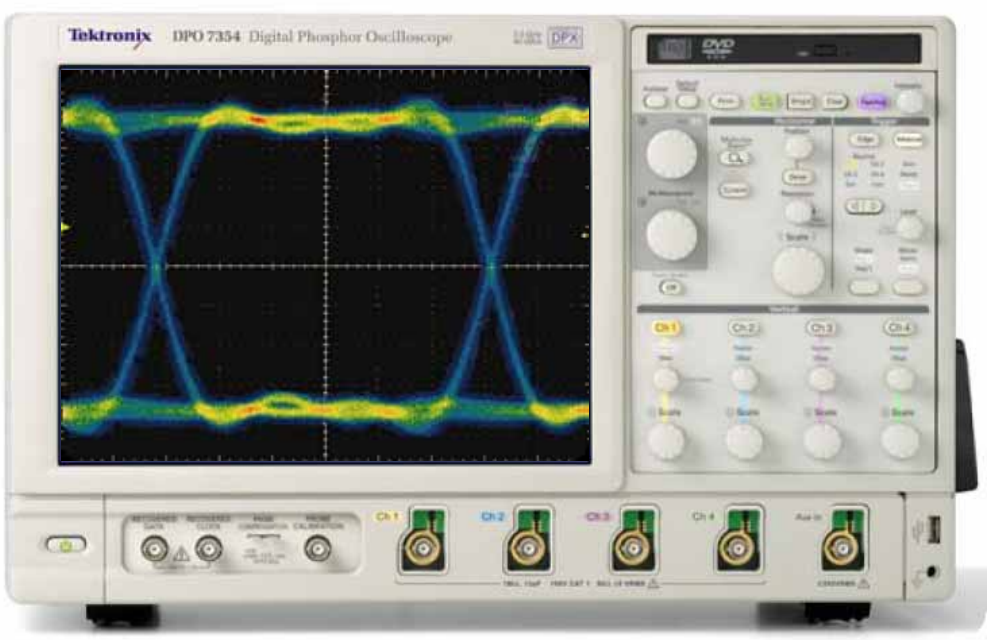
- 100 MHz to 500 MHz
- 5 M record length
- Wave Inspector
- Supported serial buses:
  - I^2C
  - SPI
  - CAN
  - LIN
  - RS-232/422/485
  - I^2S/LJ/RJ/TDM

MSO/DPO4000 Series

- 350 MHz to 1 GHz
- 10 M record length
- Wave Inspector
- Supported serial buses:
  - I^2C
  - SPI
  - CAN
  - LIN
  - FlexRay
  - RS-232/422/485
  - I^2S/LJ/RJ/TDM

In-Depth Analysis of Serial Buses with the DPO7000 Series

Automated Decode, Trigger, Search and Eye Diagram Analysis



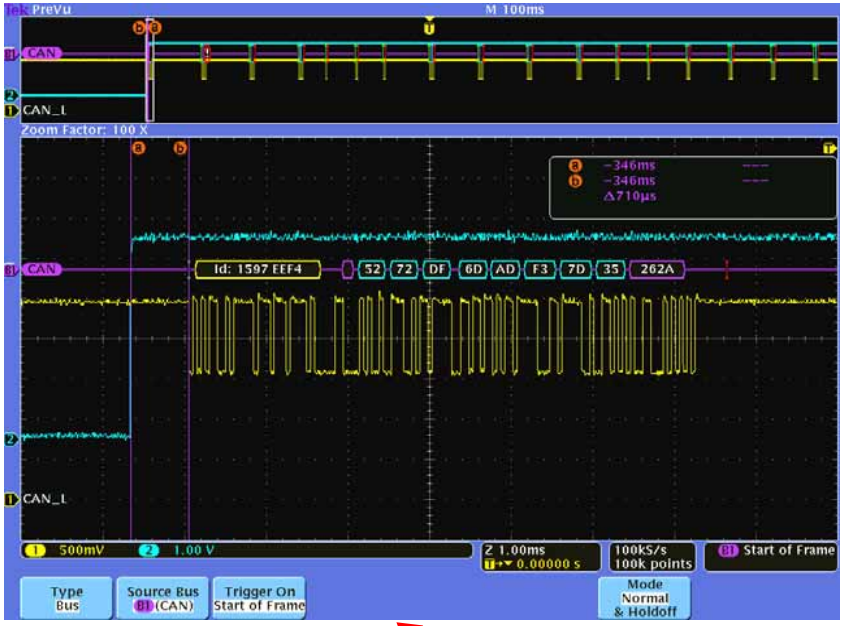
DPO7000 Series

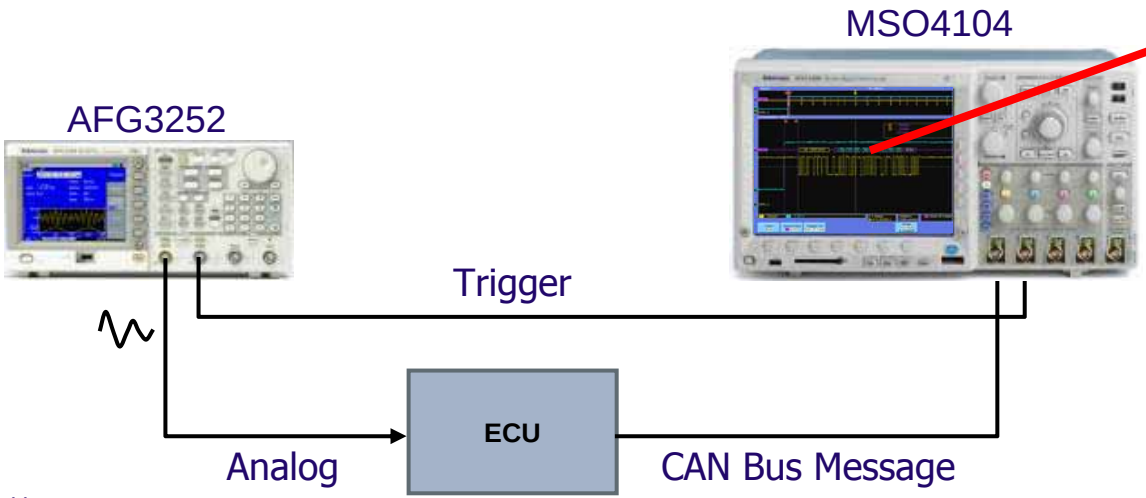
- 500 MHz to 3.5 GHz
- Up to 400 M record length
- CAN propagation delay and oscillator tolerance
- CAN bus traffic monitoring
- Supported serial buses:
  - I^2C
  - SPI
  - CAN
  - LIN
  - FlexRay
  - UART/RS-232

Tektronix Signal Generators

Evaluation of CAN Bus ECU Performance

- Signal Generator: AFG3252
  - Replicates Sensor's Signal
- Oscilloscope: MSO4104
  - Validates ECU's CAN bus message
  - Measures latency of sensor/ECU chain



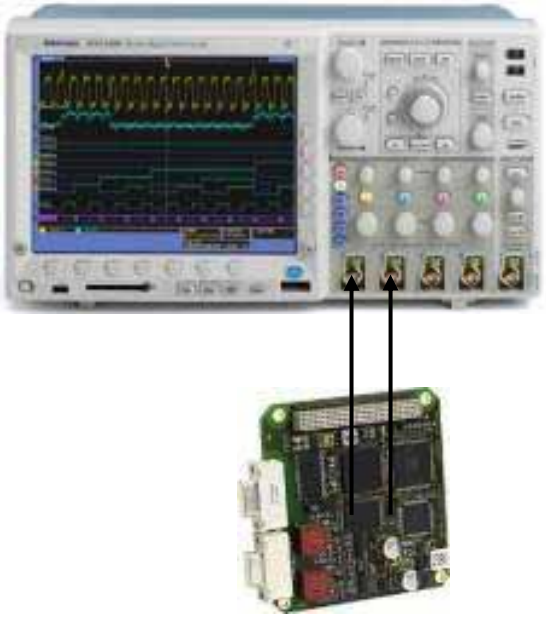


Tektronix Innovation Forum

Tektronix Signal Generators

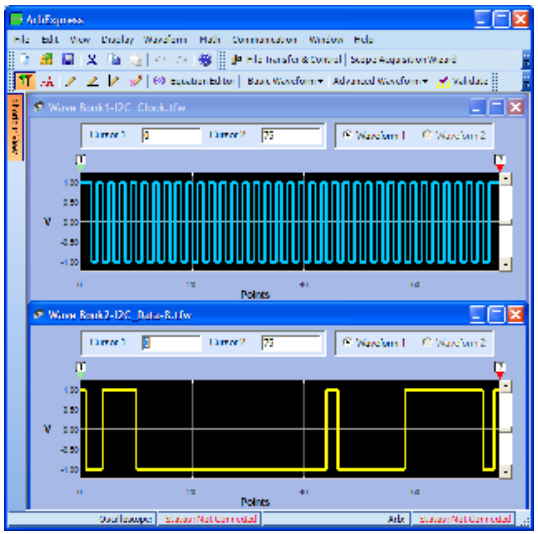
Creating Low-Speed Serial Signals

MSO4104 to Capture

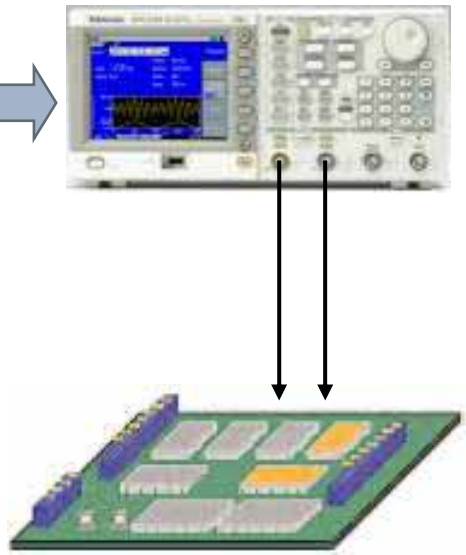


Low-Speed Serial Transmitter

ArbExpress to Import, Edit, Save, and Export



AFG3252 to Generate

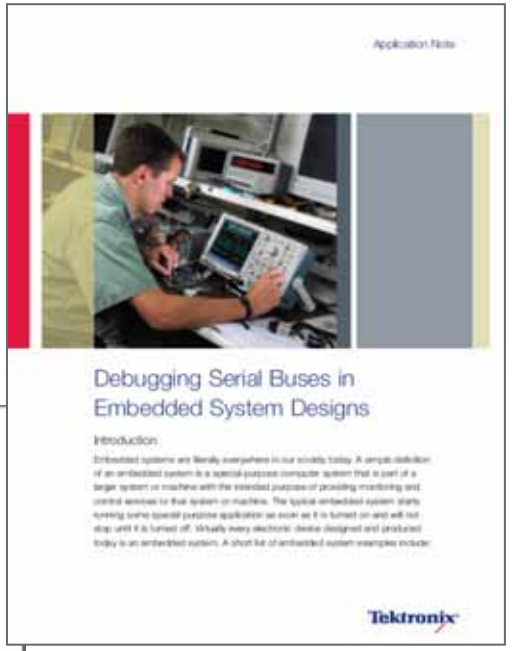


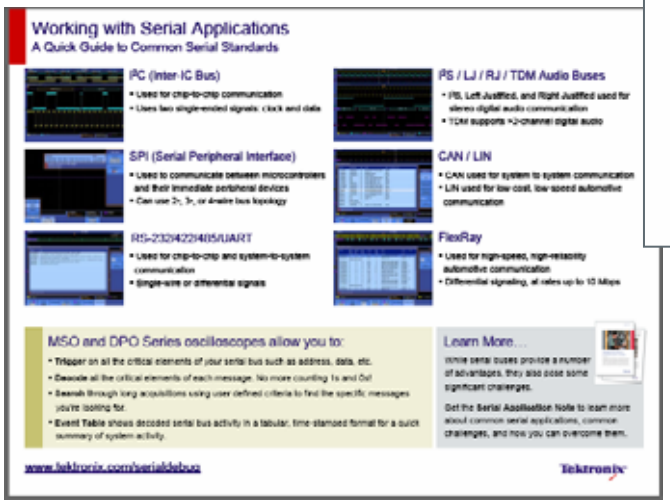
Slave Device Under Test

Tektronix Innovation Forum

Serial Bus Analysis Information

- Application notes
- Fact sheets
- Recommended test equipment
- Webinars





Visit www.tektronix.com/serialdebug

Thank You for Attending

Tektronix
Innovation Forum