Debugging Serial Parallel Buses 
In Embedded System Design

With Tektronix Oscilloscopes

---

**Serial and Parallel Busses**

- **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 mixed signal oscilloscope or a logic analyzer

- **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
Debugging Parallel Buses

- State and Pattern triggering is straightforward with mixed signal oscilloscope
- Simple decode, as each bit stream of the Bus is transmitted on distinct signal-vias
- Order of Signals when probing is relevant

Trigger on 1001 0010
Hex 92

Most Significant Bit

Least Significant Bit

1001 0000 BIN or 90 Hex
Debug of Parallel Buses

- Cursor readouts for automatic decoding of individual points on a parallel bus.

- Parallel bus decoding for automatic decoding of all points on a parallel bus.
Debug of Parallel – Setup and Hold Violations

- Setup and Hold violations
  - Metastability of the flip-flop
  - Unpredictable output

- Setup and Hold Triggering
  - Simplifies debug and Analysis
  - Available on the MSO Series

Serial Buses
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 I2C 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.*

Common Serial Buses

- **I2C**
- **SPI**
- **RS-232 (RS-422, RS-485, UART)**
- **CAN, LIN, FlexRay**
- **USB**
- **Audio (I²S, LJ, RJ, TDM)**
I^2C (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 I^2C device can be attached to the bus
- Data rates:
  - Standard Mode (100 kbps)
  - Fast Mode (400 kbps)
  - High Speed Mode (3.4 Mbps)

I^2C 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
USB 2.0 (Universal Serial Bus)

- Used for system-to-system and chip-to-chip communication
- Data rates:
  - Low Speed Mode (1.5 Mbps)
  - Full Speed Mode (12 Mbps)
  - High Speed Mode (480 Mbps)
- Configured as one host controller with up to 127 devices
  - Host controller is the only master and controls all bus traffic
  - Host initiates all communications to devices; devices cannot interrupt
- Bus speed is limited by the slowest device connected

Electrical Configuration

- USB 2.0 has four wires: two for power, two for data
- A twisted differential pair (D+ and D-) is used for bi-directional, half-duplex communications

<table>
<thead>
<tr>
<th>USB Speed</th>
<th>Low State</th>
<th>High State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Speed</td>
<td>&lt;0.3 V</td>
<td>&gt;2.8 V</td>
</tr>
<tr>
<td>Full Speed</td>
<td>&lt;0.3 V</td>
<td>&gt;2.8 V</td>
</tr>
<tr>
<td>High Speed</td>
<td>0 V ± 10%</td>
<td>400 mV ± 10%</td>
</tr>
</tbody>
</table>
Packets and Bytes

- Packets are the fundamental elements of USB communications
  - Start of Frame
  - Token
  - Data
  - Handshake
  - Special

- Each packet contains:
  - Synchronization field
  - Packet identifier (PID)
  - Additional fields, depending on packet type
  - End of Packet

<table>
<thead>
<tr>
<th>PID Type</th>
<th>PID Name</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token</td>
<td>OUT</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>1001</td>
</tr>
<tr>
<td></td>
<td>SOF</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td>SETUP</td>
<td>1101</td>
</tr>
<tr>
<td>Data</td>
<td>DATA0</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td>DATA1</td>
<td>1011</td>
</tr>
<tr>
<td></td>
<td>DATA2</td>
<td>0111</td>
</tr>
<tr>
<td></td>
<td>MDATA</td>
<td>1111</td>
</tr>
<tr>
<td>Handshake</td>
<td>ACK</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td>NAK</td>
<td>1010</td>
</tr>
<tr>
<td></td>
<td>STALL</td>
<td>1110</td>
</tr>
<tr>
<td></td>
<td>NYET</td>
<td>0110</td>
</tr>
<tr>
<td>Special</td>
<td>PRE</td>
<td>1100</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>1100</td>
</tr>
<tr>
<td></td>
<td>SPLIT</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>PING</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td>0000</td>
</tr>
</tbody>
</table>

Debugging Serial Buses
Decoding Serial Data
Example - I²C Manual Decoding

Automated Decode with Tektronix Oscilloscopes

Start
Address
Data
Stop

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

Displayed in hex or binary
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

Serial Data Triggering
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
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.

![Diagram of thermal management system](image)

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.
Serial Data Analysis
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

Serial Data Analysis
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
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

Today’s Hands-On Labs
Today’s Hands-On Labs

✓ Parallel Lab
  - Parallel Signal Acquisition
  - Triggering on Parallel Bus Values

✓ I2C Serial Lab
  - Acquisition & Bus Decode
  - Timing Measurements
  - I2C Bus Triggering, Search & Navigation

✓ SPI Serial Lab
  - Acquisition & Bus Decode
  - Timing Measurements
  - SPI Bus Triggering, Search & Navigation
Debugging Parallel and Serial Buses
Hands-on Labs to Help You Verify & Debug Your Design
## Table of Contents

**Parallel Lab** ........................................................................................................................................................... 3
- Introduction .......................................................................................................................................................... 3
- Lab Setup .......................................................................................................................................................... 3
- Parallel Signal Acquisition ................................................................................................................................. 4
- Triggering on Parallel Bus Values ..................................................................................................................... 6

**I²C Serial Lab** ..................................................................................................................................................... 7
- Introduction ........................................................................................................................................................ 7
- I²C Introduction ............................................................................................................................................... 8
- Lab Setup ........................................................................................................................................................ 9
- Acquisition of I²C Signals ................................................................................................................................. 9
- I²C Bus Decode ............................................................................................................................................... 11
- I²C Bus Timing Measurements ....................................................................................................................... 14
- I²C Bus Triggering ......................................................................................................................................... 15
- I²C Bus Searching & Navigation ..................................................................................................................... 16

**SPI Serial Lab** ..................................................................................................................................................... 17
- Introduction ...................................................................................................................................................... 17
- SPI Introduction ............................................................................................................................................ 17
- Lab Setup ...................................................................................................................................................... 18
- Acquisition of SPI Signals ............................................................................................................................... 18
- SPI Bus Decode ........................................................................................................................................... 20
- SPI Bus Timing Measurements ....................................................................................................................... 23
- SPI Bus Triggering ......................................................................................................................................... 24
- SPI Bus Searching & Navigation ..................................................................................................................... 25
# Parallel Lab

## Introduction

Various devices in embedded designs have traditionally communicated with each other and the outside world using wide parallel buses. Even though many designs today use serial buses for communication, parallel buses are still common.

With a parallel architecture, each component of the bus has its own signal path. There may be multiple address lines, multiple data lines, a clock line and various other control signals. Address or data values sent over the bus are transferred at the same time over all the parallel lines.

In this lab, you will acquire signals from a parallel bus then decode the signals to determine what data is being communicated over the bus.

## Lab Setup

This lab uses a counter output as a parallel bus.

<table>
<thead>
<tr>
<th>Procedure:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify the USB cable is plugged in to the MSO2024 Mixed Signal Oscilloscope and the DPO Demo 2 board.</td>
</tr>
<tr>
<td>Verify the demo board’s green POWER LED is on.</td>
</tr>
<tr>
<td>Connect the P6316 D0-D7 digital probe to the J924 counter output connector, being careful to align the colored label with the signals on the right side of the connector.</td>
</tr>
</tbody>
</table>
Parallel Signal Acquisition

In this section, you will acquire the signals of a counter's parallel bus. This bus has eight signals. All signals are connected to the digital channels (D0 through D7) of the MSO2024 oscilloscope.

Now, you will set up the oscilloscope to trigger on the D0 channel allowing you to acquire a stable display on the oscilloscope.

The cursor readouts provide automatic decoding of individual points on a parallel bus:

Notice that the cursor readout in the upper right hand of the display shows the data pattern of the parallel bus at the location of the cursor.

**Challenge:** Move the cursors to different points on the bus and confirm that the cursor readout is correct.
Parallel bus decoding provides automatic decoding of all points on a parallel bus:

- Set the **Horizontal Scale** to 1 µs/div.
- Press the purple front panel B1 button to enable a parallel bus.
- Press the **Define Inputs** bottom bezel button.
- Using the **Multipurpose a** control, set the **Number of Data Bits** to 8.
- Press the **Menu Off** button once to clear the side menu.
- Using the **Multipurpose a** control, position the decoded bus waveform in the top half of the display.
- Press the **Menu Off** button once to clear the menus.
- Press the front panel **Single** button to acquire one waveform.
Triggering on Parallel Bus Values

In the previous section, the oscilloscope was triggering on a rising edge of the D0 signal. The oscilloscope can also be set up to trigger on parallel bus values:

- **Procedure:**
  - Press the Trigger Menu button.
  - Press the Type bottom bezel button.
  - Turn the Multipurpose a control fully counter-clockwise to select Bus triggering.
  - Press the Data bottom bezel button.
  - Turn the Multipurpose a control fully clockwise to select the hex data word.
  - Using the Multipurpose b control, select the value 10h.
  - Press the front panel Single button.
  - Press the Menu Off button once to clear the menus.
  - Press the front panel Search button.
  - Press the Search bottom bezel button.
  - Press the Search side bezel button until On is selected.
  - Press the Search Type bottom bezel button.
  - Turn the Multipurpose a control fully counter-clockwise to select Bus searching.
  - Press the Data bottom bezel button.
  - Using the Multipurpose a control, highlight the most significant digit of the Hex value.
  - Using the Multipurpose b control, select the value X0h.
  - Press the Menu Off button twice to clear the menus.

In this case, searching on a parallel bus value X0 hex instantly found 21 matching occurrences, three of which are shown at the center of the display:
**I²C Serial Lab**

<table>
<thead>
<tr>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded systems can contain many different types of devices including microcontrollers, microprocessors, DSPs, RAM, EPROMs, FPGAs, A/Ds, D/As and I/O. These various devices must communicate with each other and the outside world using data buses. Traditionally, parallel buses were used; however, in an effort to reduce required board space, power requirements, cost and complexity, serial buses have become quite common in today’s embedded systems.</td>
</tr>
<tr>
<td>In this lab, you will explore the different features of one of the most common serial buses - I²C. You will also learn how to debug a serial bus using an oscilloscope with decode, trigger and search features for serial protocol.</td>
</tr>
</tbody>
</table>
I²C Introduction

I²C (Inter-Integrated Circuit) bus, developed in the early 1980s by Philips, has become a worldwide standard for communications between integrated circuits in a system. This simple two-wire design has found its way into a wide variety of chips and is found in many embedded designs today. I²C uses bi-directional serial clock and data lines and supports three bit rates: 100 kbps standard mode, 400 kbps fast mode and 3.4 Mbps high speed mode. Data and clock are sent from the master and the data is clocked on the rising edge of SCLK. I²C supports multiple masters and slaves on the bus, but only one master may be active at any one time while slaves can transmit or receive data to the master. Each device is recognized by a unique address and can operate as either a transmitter or receiver, depending on the function of the device.

![I²C Quick Reference Guide](image)

- **Start**: A HIGH to LOW transition on the SDA line while SCL is HIGH.
- **Stop**: A LOW to HIGH transition on the SDA line while SCL is HIGH.
- **Read**: A HIGH to LOW transition on the SDA line while SCL is LOW.
- **Write**: A HIGH to LOW transition on the SDA line while SCL is HIGH.
- **Missing ACK**: A LOW to HIGH transition on the SDA line while SCL is HIGH.
- **ACK**: A LOW to HIGH transition on the SDA line while SCL is HIGH.
- **0-8 bytes**: A LOW to HIGH transition on the SDA line while SCL is HIGH.

---

<table>
<thead>
<tr>
<th>Start/Stop</th>
<th>Addr</th>
<th>R/W</th>
<th>Ack</th>
<th>Data</th>
<th>Ack</th>
<th>Data</th>
<th>Ack</th>
<th>Data</th>
<th>Ack</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 bit</td>
<td>7-bits</td>
<td>1-bit</td>
<td>1-bit</td>
<td>8-bits</td>
<td>1-bit</td>
<td>8-bits</td>
<td>1-bit</td>
<td>0-8 bytes</td>
<td>8-bits</td>
<td>1-bit</td>
</tr>
</tbody>
</table>
### Lab Setup

This lab uses I2C_CLK and I2C_DATA connections located on the J919 black square pins of the DPO Demo 2 board.

**Procedure:**

- Verify the USB cable is plugged into the MSO2024 Mixed Signal Oscilloscope and the DPO Demo 2 board.
- Verify the green **POWER** LED is on.
- Connect the P6316 D8-D15 digital probe to the J919 black square pins. Be sure the **Ground** side of the probe is positioned on the left side of the connector.

### Acquisition of I²C Signals

In this section, you will acquire the signals of an I²C bus. This bus has two signals: I2C_CLK and I2C_Data. Both signals are connected to the digital channels (D8 and D9) of the MSO2024 oscilloscope.

**Procedure:**

- Press the front panel **Default Setup** button.
- Remove channel one waveform from the display by pressing the yellow **Vertical channel one** button twice.
- Press the **blue digital channels D15-D0** button.
- Turn off digital channel D0 by pressing **D15-D0 On/Off** bottom bezel button and the **Display On Off** side bezel button.
- Turn on D8 by using the **Multipurpose** a control to select D8 and then turn D8 on with the **Display On Off** side bezel button.
- Turn on D9 with the **Display On Off** side bezel button.
Now, you will set up the oscilloscope to trigger on the D8 channel allowing you to acquire a stable display on the oscilloscope.

- Press the front panel **Trigger Menu** button.
- Change the trigger source to D8 with the **Source** bottom bezel button and the **Multipurpose** a control.
- Press **Menu Off** twice to remove the menus.
- Set the **Horizontal Scale** to 40 µs/div.
- Press the front panel **Single** acquisition button a few times to see the I²C clock and data waveforms.
I²C Bus Decode

In this section you will learn how to automatically decode I²C packet content. Setting up an I²C bus decode display takes only a few simple steps with the MSO2024’s automated decode feature. After turning on a bus with a front panel button, the menus guide you through the setup in the left-to-right order across the bottom bezel buttons. The I²C bus selection menu looks like this:

The analog signals can be used as a source for the I²C bus, as well as the digital inputs on the MSO2024. In these steps, you’ll define the I²C Clock line (SCLK) as channel D8 and the I²C Data line (SDA) as channel D9.

Here is an example of I²C bus decode. Your I²C bus decode will be different depending upon the I²C packets that you are viewing.

Procedure:
- Press the purple Bus B1 button.
- Move the purple bus up with the Multipurpose a control so that the bus does not cover the I²C waveforms.
- Press the Bus B1 bottom bezel button.
- Select I²C using the Multipurpose a control.
- Press the Define Inputs bottom bezel button.
- Set SCLK Input to channel D8 using the Multipurpose a control.
- Set SDA Input to channel D9 using the Multipurpose b control.
- Press front panel Menu Off button once to remove the bottom menu.
- I²C bus is decoded.
- Press the front panel Single acquisition button a few times to see the different I²C packet decodes.
The MSO2024’s Wave Inspector lets you zoom in to see more detail around your decoded bus.

Looking at your I²C packets, note the following:

- The **green bar** symbol represents the start of packet. Start is a high to low transition on the data signal while the clock is high.
- Address packets are shown in yellow boxes. Addresses are 7-bits or 10-bits long.
- [R] Indicates a read and [W] indicates a write.
- Data packets are shown in cyan boxes.
- After each packet, the device produces an acknowledge. The MSO2024 only displays missing acknowledges. Missing Acknowledges are indicated by the symbol (!).
- The **red bar** symbol represents the end of the packet. The end of the packet is a low to high transition on the data signal while the clock is high.

**Procedure:**

- Using the inner **Wave Inspector Zoom** control, zoom in on one of the packets.
- Turn off zoom with the front panel zoom button (magnifying glass icon).
- Set the **Horizontal Scale** to 400 µs/div to acquire more I²C data.
- Press the front panel **Single** button to acquire a waveform.
- Using the **Wave Inspector Zoom and Pan** controls, zoom in to see packet details and pan through your decoded bus to see other packets in your acquisition.
In the last section, the default I²C decoder display format is a hexadecimal display of the ID and Parity, and hexadecimal for the Data and Checksum values. These values can also be displayed in binary.

**Procedure:**

- Press the **purple Bus B1** button.
- Press the **Bus Display** bottom bezel button.
- Notice the Hex and Binary display settings. Try the Binary display settings
- Set the display back to Hex before proceeding.

For many users who are not electrical engineers, the waveform displays may not be familiar. For them, the Event Table format is preferable. This format can also be saved in CSV format and analyzed off-line.

**Procedure:**

- Press the **Event Table** bottom bezel button.
- Press the **Event Table** side bezel button until it is set to **On**.
- Notice that all of the decoded packets in your signal are listed in the table.
- Using the **Event Table** side bezel button, select **Off**.
- Press front panel **Menu Off** button twice to remove the menus.
I²C Bus Timing Measurements

The I²C serial cursors provide another tool to bridge between the serial waveform and the decoded values, showing absolute timing relative to the trigger and bus values, and relative timing between bus events.

In this section, you will measure the time from the start of the address to the start of the first data packet.

**Procedure:**

- Using the inner *Wave Inspector Zoom* control, zoom in on one of the I²C messages.
- Press the *Cursors* front panel button once to turn on vertical bar cursors.
- Using the *Multipurpose a* control, place the a cursor on the start of the address.
- The a cursor time and bus value readout is in the upper right corner of the display.
- Using the *Multipurpose b* control, place the b cursor on the start of the first data packet.
- The b cursor time and bus value readout is in the upper right corner of the display.
- The cursor readout displays the time difference between the start of the address and the start of the first data packet.
- Turn off cursors by pressing the *Cursors* front panel button.

**Challenge:** Measure the timing between the “Start of Packet” and the “Start of the Address” on at least two packets. What’s the typical value?
I²C Bus Triggering

When debugging a system, you often want to capture the state of some key signals when a certain event occurs. One key event may be the transmission of specific content over the I²C bus.

The MSO2024 Mixed Signal Oscilloscope triggers on:

- Start
- Repeat Start
- Stop
- Missing Ack
- Address
- Data (1 to 5 bytes)
- Address & Data

By following this simple procedure, you can easily trigger the scope on a specified serial pattern on an I²C bus, capturing each occurrence.

**Procedure:**

- Press the **Trigger Menu** front panel button.
- Press the **Type** bottom bezel button.
- Select **Bus** triggering using the **Multipurpose a** control. The Bus selection is at the bottom of the menu and will scroll up.
- Press the **Trigger On** bottom bezel button.
- Using the **Multipurpose a** control, select **Data**.
- Press the **Data** bottom bezel button.
- Press the **Data** side bezel button.
- Use the **Multipurpose a and b** controls to set Hex data to 14 hex.
- Press front panel **Menu Off** button twice to remove the menus.
- Press the front panel **Single** acquisition button.
- Adjust the **Wave Inspector Pan and Zoom** controls as needed to view the I²C data 14 hex trigger packet. (Hint: The Trigger event is marked by an orange triangle.)

**Challenge:** Why would you want to trigger on a specific address? Data packet? Missing Acknowledgement?
I²C Bus Searching & Navigation

A long-record-length oscilloscope, such as the MSO2024, captures long time spans at high timing resolution. However, it is difficult to find specific I²C packet characteristics in a long acquisition.

Automated search provides an easy way to find all occurrences of a specific I²C characteristic, such as when the I²C data packet value is 14 hex. In this section, you will search your I²C acquisition to find every instance when the data packet has a value of 14 hex.

Wave Inspector search finds all occurrences of a specified event. In the below display, Wave Inspector found 5 packets with data 14 hex. White triangular marks appear at the top of the display at each search event.

Pressing the front panel Search arrow buttons quickly jumps you to the next I²C data 14 hex.

Procedure:

- Turn off zoom with the front panel zoom button (magnifying glass icon).
- Set the Horizontal Scale to 4 ms/div.
- Press the Horizontal Acquire front panel button.
- Press the Record Length bottom bezel button.
- Press the 1.25M points side bezel button.
- Press the front panel Single acquisition button to acquire one waveform.
- Press the Search front panel button.
- Press the Search bottom bezel button.
- Press the Search side bezel button until Search is turned On.
- Press the Copy Trigger Settings to Search side bezel button.
- Adjust the Wave Inspector Pan and Zoom controls as needed to view the I²C data 14 hex trigger packet.
- Press front panel Menu Off button twice to remove the menus.
- Navigate quickly between marked occurrences by using the front panel Search ← and → buttons.
# SPI Serial Lab

## Introduction

In this lab, you will explore the different features of another common serial bus - SPI. You will be able to practice debugging a serial bus using an oscilloscope with decode, trigger and search features for serial protocol.

## SPI Introduction

SPI (Serial Peripheral Interface) bus is a 4-wire serial communications interface used primarily in synchronous serial communication for both processors and peripherals. SPI is an interface standard defined by Motorola for use on their microcontrollers. Due to the popularity of the bus other manufacturers have adopted the standard, making a wide variety of parts available in the market.

SPI uses a synchronous clock which shifts serial data into and out of the microcontroller in blocks of 8 bits. The SPI bus is a master/slave interface. Whenever two devices communicate, one is referred to as the "master" and the other as the "slave". The master drives the serial clock. When using SPI, data is simultaneously transmitted and received, making it a full-duplex protocol.

SPI is a flexible interface, supporting several different circuit topologies. In general, the master provides a clock to all slaves. When MOSI (Master Out Slave In), SCLK (Serial Clock), and SS (Slave Select active low) are used, it is called the “3-wire” SPI connection. With the single-master, multi-slave topology, the master provides data (MOSI) directly to each slave and controls them separately with slave select (SS) signals. This is the configuration that will be used in this lab.

## SPI Quick Reference Guide

![SPI Quick Reference Guide](image-url)
Lab Setup

This lab uses SPI_SS, SPI_MOSI and SPI_SCK connections located on the J919 black square pins of the DPO Demo 2 board.

Procedure:
- Verify the USB cable is plugged into the MSO2024 Mixed Signal Oscilloscope and the DPO Demo 2 board.
- Verify the green POWER LED is on.
- Verify the P6316 D8-D15 digital probe is connected to the J919 black square pins. Be sure the Ground side of the probe is positioned on the left side of the connector.

Acquisition of SPI Signals

In this section, you will acquire the signals of a SPI bus. This bus has three signals: SPI_SS, SPI_MOSI, and SPI_SCLK. The three signals are connected to the digital channels (D10, D11 and D12) of the MSO2024 oscilloscope.

Procedure:
- Press Default Setup button.
- Remove the yellow waveform from the display by pressing the yellow Vertical channel one button twice.
- Press the blue digital channels D15-D0 button.
- Turn off digital channel D0 by pressing D15-D0 On/Off bottom bezel button and the Display On Off side bezel button.
- Turn on D10 by using the Multipurpose a control to select D10 and then turn D10 on with the Display On Off side bezel button.
- Turn on D11 with the Display On Off side bezel button.
- Turn on D12 with the Display On Off side bezel button.
Now, you will set up the oscilloscope to trigger on the D10 channel allowing you to acquire a stable display on the oscilloscope.

- Press the front panel **Trigger Menu** button.
- Change the trigger source to D10 (SS) with the **Source** bottom bezel button and the **Multipurpose** control.
- Press the trigger **Slope** bottom bezel button to select the falling edge of D10 as the trigger point.
- Press **Menu Off** twice to remove the menus.
- Set the **Horizontal Scale** to 40 µs/div.
- Press the front panel **Single** acquisition button a few times to see the SPI clock, data and SS waveforms.
In this section you will learn how to automatically decode SPI packet content. Setting up a SPI bus decode display takes only a few simple steps with the MSO2024. After turning on a bus with a front panel button, the menus guide you through the setup in the left-to-right order across the bottom bezel buttons. The SPI bus selection menu looks like this:

The analog signals can be used as a source for the SPI bus, as well as the digital inputs on the MSO2024. In these steps, you’ll define the SPI Slave Select line (SS) as channel D10, the SPI Clock line (SCLK) as channel D12, and the SPI Master Out Slave In line (MOSI) as channel D11.

**Procedure:**
- Press the **purple Bus B1** button.
- Move the purple bus up with the **Multipurpose a** control so that the bus does not cover the SPI waveforms.
- Press the **Bus B1** bottom bezel button.
- Select SPI using the **Multipurpose a** control.
- Press the **Define Inputs** bottom bezel button.
- Set SCLK Input to channel D12 using the **Multipurpose a** control.
- Press the **SS Input** side bezel button.
- Set SS Input to channel D10 using the **Multipurpose a** control.
- Press the **MOSI Input** side bezel button.
- Set MOSI Input to channel D11 using the **Multipurpose a** control.
- Press front panel **Menu Off** button twice to remove the menus.
- SPI bus is decoded.
- Press the front panel **Single** acquisition button a few times to see the different SPI packet decodes.
Below is an example of SPI bus decode. Your SPI bus decode will be different depending upon the SPI packets that you are viewing.

Looking at your SPI packets, note the following:

- The **green bar** symbol represents the start of packet. Start is a high to low transition on the SS signal.
- MOSI data packets are shown in **cyan boxes**.
- The **red bar** symbol represents the end of the packet. This occurs at the low to high transition on the SS signal.

**Procedure:**

- Using the inner **Wave Inspector Zoom** control, zoom in on one of the packets.
- Turn off zoom with the front panel zoom button (**magnifying glass icon**).
- Set the **Horizontal Scale** to 400 µs/div to acquire more SPI data.
- Press the front panel **Single** button to acquire a waveform.
- Using the **Wave Inspector Zoom and Pan** controls, zoom in to see packet details and pan through your decoded bus to see other packets in your acquisition.
In the last section, the default SPI decoder display format is hexadecimal display of the MOSI data. The MOSI data values can also be displayed in binary.

**Procedure:**
- Press the **purple Bus B1** button.
- Press the **Bus Display** bottom bezel button.
- Notice the Hex and Binary display settings. Try the Binary display settings.
- Set the display back to Hex before proceeding.

For many users who are not electrical engineers, the waveform displays may not be familiar. For them, the Event Table format is preferable. This format can also be saved in CSV format and analyzed off-line.

- Press the **Event Table** bottom bezel button.
- Press the **Event Table** side bezel button until it is set to **On**.
- Notice that all of the decoded packets in your signal are listed in the table.
- Using the **Event Table** side bezel button, select **Off**.
- Press front panel **Menu Off** button twice to remove the menus.
**SPI Bus Timing Measurements**

The SPI serial cursors provide another tool to bridge between the serial waveform and the decoded values, showing absolute timing relative to the trigger and bus values, and relative timing between bus events. In this section, you are going to measure the time from one MOSI data packet to another MOSI data packet.

Even when you are zoomed out to where you can’t read the decoded values on the display, cursor measurements enable you to make timing measurements on buses. Notice that the MOSI data packets selected by the cursors are clearly and easily displayed:

**Challenge:** Measure the timing between the “Start of Frame” and the “Start of MOSI Data” on at least two packets. What’s the typical value?

**Procedure:**

- Using the inner **Wave Inspector Zoom** control, zoom in on one of the SPI messages.
- Press the **Cursors** front panel button once to turn on vertical bar cursors.
- Using the **Multipurpose a** control, place the a cursor on one of the MOSI data packets.
- The a cursor time and bus value readout is in the upper right corner of the display.
- Using the **Multipurpose b** control, place the b cursor on another MOSI data packet.
- The b cursor time and bus value readout is in the upper right corner of the display.
- The cursor readout displays the time difference between the MOSI data packets.
- Turn off zoom with the front panel zoom button (**magnifying glass icon**).
- Slowly move the cursors to other MOSI data packets. You can see the cursor MOSI data packets and their timing without being able to clearly see the waveforms or the decoded values.
- Turn off cursors by pressing the **Cursors** front panel button.
When debugging a system, you often want to capture the state of some key signals when a certain event occurs. One key event may be the transmission of specific content over the SPI serial bus.

The MSO2024 Mixed Signal Oscilloscope triggers on:
- SS Active or Start of Frame
- MOSI data
- MISO or MOSI & MISO (if MISO is set up)

By following this simple procedure, you can easily trigger the scope on a specified serial pattern on a SPI signal, capturing each occurrence.

**Procedure:**
- Press the **Trigger Menu** front panel button.
- Press the **Type** bottom bezel button.
- Select **Bus** triggering using the **Multipurpose a** control. The Bus selection is at the bottom of the menu and will scroll up.
- Press the **Trigger On** bottom bezel button.
- Press the **MOSI** side bezel button.
- Press the **Data** bottom bezel button.
- Press the **SPI Data MOSI** side bezel button.
- Using the **Multipurpose a and b** controls to set Hex data to 10 hex.
- Press front panel **Menu Off** button twice to remove the menus.
- Press the front panel **Single** acquisition button.
- Adjust the **Wave Inspector Pan and Zoom** controls as needed to view the SPI data 10 hex trigger packet. (Hint: The Trigger event is marked by an orange triangle.)
SPI Bus Searching & Navigation

A long-record-length oscilloscope, such as the MSO2024, captures long time spans at high timing resolution. However, it is difficult to find specific SPI packet characteristics in a long acquisition.

Automated search provides an easy way to find all occurrences of a specific SPI characteristic, such as when the SPI data packet value is 10 hex. In this section, you will search your SPI acquisition to find every instance when the data packet has a value of 10 hex.

Wave Inspector search finds all occurrences of a specified event, such as SPI data 10 hex. In the below display, Wave Inspector found 2 packets with data 10 hex. White triangular marks appear at the top of the display at each search event. Pressing the front panel Search arrow buttons quickly jumps you to the next SPI data 10 hex.

Procedure:
- Turn off zoom with the front panel zoom button (magnifying glass icon).
- Set the Horizontal Scale to 4 ms/div.
- Press the Horizontal Acquire front panel button.
- Press the Record Length bottom bezel button.
- Press the 1.25M points side bezel button.
- Press the front panel Single acquisition button.
- Press the Search front panel button.
- Press the Search bottom bezel button.
- Press the Search side bezel button until Search is turned On.
- Press the Copy Trigger Settings to Search side bezel button.
- Adjust the Wave Inspector Pan and Zoom controls as needed to view the SPI data 10 hex trigger packet.
- Press front panel Menu Off button twice to remove the menus.
- Navigate quickly between marked occurrences by using the front panel Search ← and → buttons.