ScannerMAX

Stronger • Cooler • Faster

LabVIEW DSP Serial Communication Interface
User Manual and Documentation

July 29, 2025

Contents

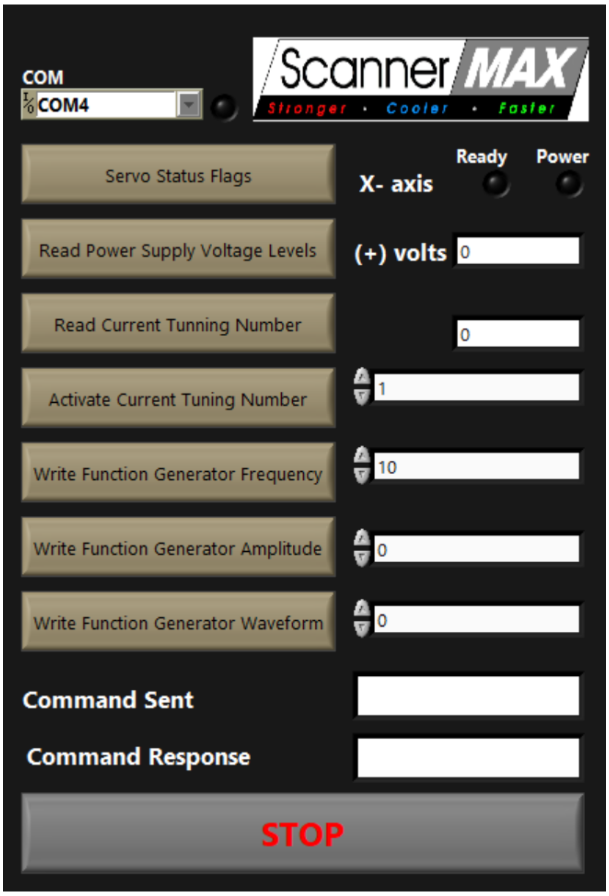
1 Introduction

1.1 Purpose

The LabVIEW Mach-DSP Serial Communication VI is designed to control and monitor a DSP device via VISA serial communication. The virtual instrument (VI) provides an interface for interacting with the device's various functions, including servo status monitoring, power supply voltage reading, tuning number configuration, and function generator control.

1.2 Key Features

- VISA serial communication interface for Mach-DSP device
- Real-time servo status monitoring (ready state and power status)
- Power supply voltage measurement
- Tuning number read/activate functionality
- Function generator control (frequency, amplitude, waveform)
- Event-driven architecture with clear command/response indicators



COM: COM4

(+) volts: 0

ScannerMAX
Stronger • Cooler • Faster
Servo Status Flags
X- axis
Ready
Power
Read Power Supply Voltage Levels
Read Current Tunning Number
0
Activate Current Tuning Number
1
Write Function Generator Frequency
10
Write Function Generator Amplitude
0
Write Function Generator Waveform
0
Command Sent
Command Response
STOP

Figure 1: LabVIEW front panel

2 System Requirements

2.1 Hardware Requirements

- Mach-DSP device with serial communication capability
- USB-to-serial adapter (if needed)
- Computer with available USB ports

2.2 Software Requirements

- LabVIEW 2018 or later
- NI-VISA drivers installed

3 Getting Started

3.1 Establishing Connection

1. Connect your Mach-DSP device to the computer using the serial port
2. Open the VI (match_dsp_serialvi.vi)
3. Select the appropriate VISA resource name (COM port)
4. The VI will automatically configure the serial port with these settings:
    - Baud rate: 256000

3.2 Monitoring Servo Status

1. Click the "Servo Status Flags" button (set to True)
2. Observe the status indicators:
    - Ready: Indicates DSP ready status
    - Power: Indicates power status
3. View the raw command and response in the corresponding indicators

3.3 Reading Power Supply Voltages

1. Click the "Read Power Supply Voltage Levels" button (set to True)
2. View the response in the "command response" indicator
3. The voltage value will be encoded in the response string

3.4 Working with Tuning Numbers

1. To read current tuning number:
    - Click "Read Tunning Number" button (set to True)
    - View response in the command response indicator
2. To activate a tuning number:
    - Click "Activate Current Tuning Number" button (set to True)
    - The default tuning number (1) will be sent

3.5 Controlling Function Generator

1. Set frequency:
    - Click "Write Function Generator Frequency" button (set to True)
    - The default frequency (10) will be sent
2. Set amplitude:
    - Click "Write Function Generator Amplitude" button (set to True)
    - The default amplitude (0) will be sent
3. Set waveform:
    - Click "Write Function Generator Waveform" button (set to True)
    - The default waveform (0) will be sent

4 Technical Details

4.1 Serial Communication Protocol

The VI communicates with the DSP using a binary protocol. Each command consists of a specific byte sequence, and responses are also byte sequences.

4.2 Command Reference

Command	Response Pattern	Description
`0x80 0x00 0x00 0x00`	`0x55 0x00 X X`	Read servo status flags (X contains flag bits)
`0x80 0x01 0x00 0x00`	`0x55 0x01 X X`	Read power supply voltage (X = voltage data)
`0x80 0xF1 0x80 0x00`	`0x55 0xF1 X X`	Read tuning number (X = current tuning number)
`0xC0 0xF1 0x80 0x0N`	`0xAA 0xF1 X X`	Activate tuning number (N = new tuning number)
`0xC0 0x1D X X X X`	`0xAA 0x1D X X`	Set function generator frequency
`0xC0 0x1E X X X X`	`0xAA 0x1E X X`	Set function generator amplitude
`0xC0 0x1F X X X X`	`0xAA 0x1F X X`	Set function generator waveform

Table 1: Serial command reference (bytes in hexadecimal)

4.3 Response Processing

The VI processes responses as follows:

1. Converts response bytes to hexadecimal string
2. Matches pattern to determine response type
3. Extracts relevant data bytes
4. Converts to appropriate data type (numeric or Boolean)
5. Updates indicators accordingly

4.4 Data Extraction Details

- **Power Supply Voltage**:
  - Extracts bytes 4-7 from response
  - Converts hex string to numeric value
  - Scales by dividing by 100
- **Tuning Number**:
  - Extracts byte 7 from response
  - Converts hex string to integer
- **Status Flags**:
  - Extracts bytes 4-7 from response
  - Converts to integer then to Boolean array
  - Checks bits 9 (Ready) and 10 (Power)

5 Block Diagram Explanation

The VI follows this workflow:

1. Configures serial port with VISA Open and VISA Configure
2. Entries main loop:
   - Checks which controls are activated

  - Builds appropriate command string
  - Sends command with VISA Write
  - Reads response with VISA Read
  - Parses response and updates indicators

3. On exit, closes VISA session

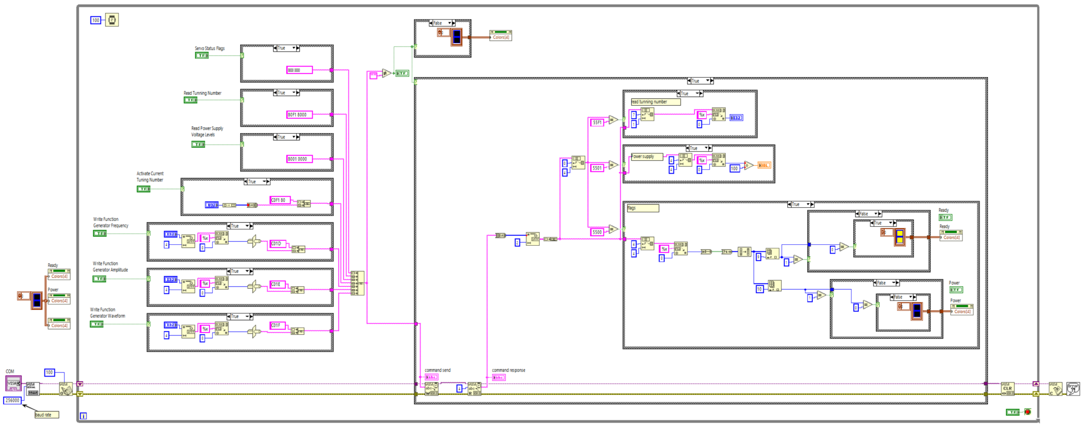


Figure 2: Block diagram

6 Troubleshooting

Issue	Solution
No VISA resources available	Check physical connection, install VISA drivers
Communication fails	Verify baud rate (256000), check cable
No response from device	Verify device is powered on and functioning
Incorrect data parsing	Check response pattern matching in code

Table 2: Troubleshooting guide

7 Appendices

7.1 Version History

- 1.0: Initial LabVIEW release