Pangolin Serial Communication Interface User Manual and Documentation

July 23, 2025

Contents

1 Introduction

1.1 Purpose

The Pangolin Serial Communication Interface is a Python application designed to control and monitor the Match DSP controller via serial communication. The application provides a graphical user interface (GUI) for interacting with the controller's various functions, including servo status monitoring, power supply voltage reading, tuning number configuration, and function generator control.

1.2 Key Features

- Serial communication interface for Match DSP controller
- Real-time servo status monitoring (ready state and power supply faults)
- Power supply voltage measurement (positive and negative rails)
- Tuning number read/write functionality
- Function generator control (frequency, amplitude, waveform)
- Multi-threaded architecture for responsive UI

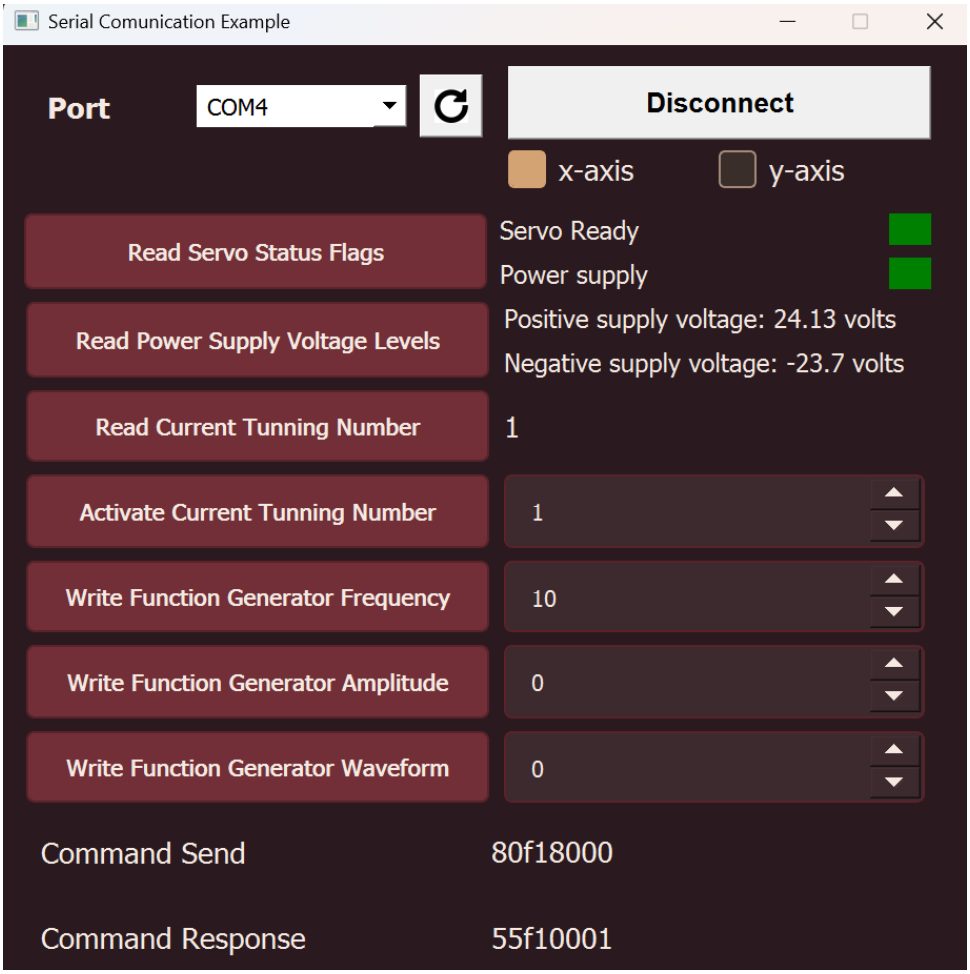

Figure 1: Main application window

2 System Requirements

2.1 Hardware Requirements

- Match DSP controller with serial communication capability

- USB-to-serial adapter (if needed)
- Computer with available USB ports

2.2 Software Requirements

- Python 3.7 or later
- Required Python packages:
  - PyQt5
  - pyserial
  - serial.tools

3 Installation Guide

3.1 Python Installation

1. Download Python from `https://www.python.org/downloads/`
2. Run the installer, checking "Add Python to PATH"
3. Verify installation by running `python --version` in Command Prompt

3.2 Library Installation

Run these commands in Command Prompt:

```
pip install pyqt5 pyserial
```

3.3 Running the Application

1. Navigate to the application directory
2. Run the main script:

```
python pangolin_serial.py
```

4 Getting Started

4.1 Establishing Connection

1. Connect your Match DSP controller to the computer using the serial port (Figure 2.)
2. Open the application
3. Select the appropriate COM port from the dropdown
4. Click "Connect"
5. Verify connection by checking servo status

4.2 Monitoring Servo Status

1. Select the desired axis (X or Y)
2. Click "Get Flags" button
3. Observe the status indicators:
   - Servo Ready: Green = ready, Yellow = not ready
   - Power Supply: Green = OK, Red = fault

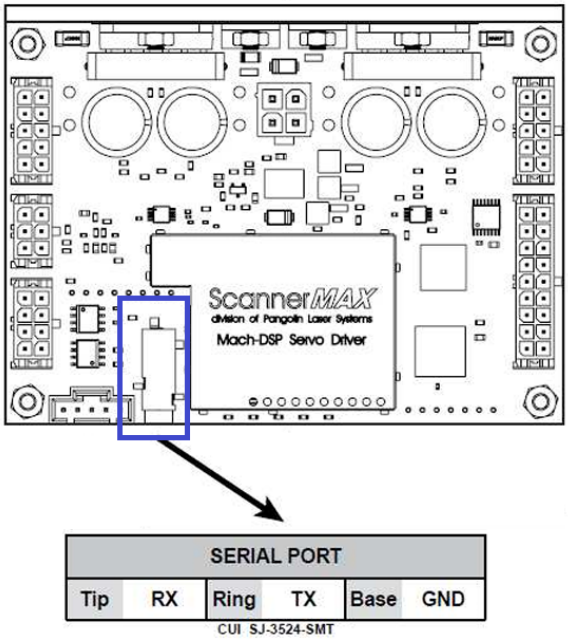

Figure 2: Match-DSP serial port

4.3 Reading Power Supply Voltages

1. Click "Power Supply" button
2. View the positive and negative supply voltages
3. Normal range: ±24V (may vary by system)

4.4 Working with Tuning Numbers

1. To read current tuning number:
   - Click "Read Tuning" button
   - View value in tuning number display
2. To write new tuning number:
   - Enter desired value in spinbox (1-4)
   - Click "Activate Tuning Number" button
   - Verify by reading the value back

4.5 Controlling Function Generator

1. Set frequency:
   - Enter desired frequency in Hz in spinbox
   - Click "Write Frequency" button
2. Set amplitude:
   - Enter desired amplitude in spinbox
   - Click "Write Amplitude" button
3. Set waveform:
   - Enter waveform code in spinbox
   - Click "Write Waveform" button

5 Technical Details

5.1 Serial Communication Protocol

The application communicates with the Match DSP using a custom hexadecimal protocol. Each command consists of an 8-character hex string, and responses are also 8-character hex strings. For a full review of the serial communication protocols of match-dsp please refer to the serial protocol manual

5.2 Command Reference

Command	Response	Description
80000000	5500XXXX	Read servo status flags (XXXX contains flag bits)
80010000	5501XXXX	Read positive power supply voltage (XXXX = voltage × 100)
80010100	5501XXXX	Read negative power supply voltage (XXXX = voltage × 100)
80f18000	55f1XXXX	Read tuning number (XXXX = current tuning number)
c0f1800N	aaf1XXXX	Write tuning number (N = new tuning number)
c01dXXXX	aa1dXXXX	Set function generator frequency
c01eXXXX	aa1eXXXX	Set function generator amplitude
c01fXXXX	aa1fXXXX	Set function generator waveform

Table 1: Serial command reference

5.3 Flag Interpretation

The servo status flags (from command 80000000) are interpreted as follows:

- Bit 1: Y-axis servo ready (1 = ready, 0 = not ready)
- Bit 2: Y-axis power supply fault (1 = fault, 0 = OK)
- Bit 9: X-axis servo ready (1 = ready, 0 = not ready)
- Bit 10: X-axis power supply fault (1 = fault, 0 = OK)

6 Troubleshooting

Issue	Solution
No COM ports available	Check physical connection, try different USB port
Connection fails	Verify baud rate (256000), check cable
No response from device	Verify device is powered on and functioning
Incorrect readings	Check axis selection matches physical configuration

Table 2: Troubleshooting guide

7 Appendices

7.1 Version History

- 1.0: Initial release