# **Official CyberBee Documentation: Client-Side Implementation Guide for Reading Pose Data via UART/USB**

## **Table of Contents**

## **Table of Contents**

1. Introduction
2. Client-Side Setup
   1. Hardware Connection
   2. Verifying Connection
3. Serial Communication API Documentation
   1. API Overview
   2. Command Structure
   3. Available Commands
4. Incoming Message Structure
5. Configuration and Startup Routine
   1. Default Setup
   2. Start Routine
   3. Set Baud Rate Routine
6. System Requirements
7. Installation Instructions
8. Troubleshooting and FAQs
9. Advanced Configuration and Customization
10. Security and Privacy Considerations
11. Performance and Optimization
12. Integration with Other Systems
13. Firmware and Software Updates
14. Technical Support and Customer Service
15. Warranty and Service Information
16. Regulatory Compliance and Certifications
17. Glossary of Terms
18. Appendices
    1. Sample Code and Scripts
    2. Compatibility Lists
    3. Change Log and Version History

## **1. Introduction**

This document serves as a guide for clients to connect and effectively read pose data transmitted over UART/USB interfaces. It is tailored for applications in robotics and similar fields. The guide includes instructions on hardware connection, utilizing a Python script for serial communication, and detailed API documentation for serial commands.

## **2. Client-Side Setup**

### **2.1 Hardware Connection**

* **Identify USB/UART Port:**
  + Connect the module to your computer using the micro USB connections on the board.
  + The port
  + number will be COMx on Windows or /dev/ttyUSBx or /dev/ttyACMx on Linux.

### **2.2 Verifying Connection**

* **Tools Required:**
  + Use PuTTY for Windows or screen for Linux.
  + Configure these tools with the correct port number and baud rate (default 115200).

## **3. Serial Communication API Documentation**

### **3.1 API Overview**

This section details the serial commands used to communicate with the hardware driver.

### **3.2 Command Structure**

* Format: <ST>;command;data;<EN>

### **3.3 Available Commands**

1. **Test Command**
   * Command: UserCommand.Test (0x20)
   * Data: 0
   * Example: <ST>;0x20;0;<EN>
2. **Change Baud Rate**
   * Command: UserCommand.ChangeBaudRate (0x30)
   * Data: Index in the baud\_rates array
   * Example: <ST>;0x30;2;<EN> (for 38400 baud rate)
3. **Set Division Rate**
   * Command: UserCommand.SetDivisionRate (0x22)
   * Data: Division rate (0-5)
   * Example: <ST>;0x22;3;<EN> (division rate 3)
4. **Additional Commands**
   * **Confirm Command:**
     + Command: UserCommand.Confirm (0x23)
     + Data: 0
     + Example: <ST>;0x23;0;<EN>

## **4. Incoming Message Structure**

Format: <ST>;message\_type;timestamp;position\_data;quaternion\_data;velocity\_data;checksum;<EN>

Components:

1. message\_type: A byte indicating the data type, e.g., 01 for ODOMETRY, 02 for POSITION.
2. timestamp: A Unix timestamp in seconds with nanosec resolution.
3. position\_data: Three floating-point numbers for X, Y, Z coordinates.
4. quaternion\_data: Four floating-point numbers representing the quaternion (W, X, Y, Z).
5. velocity\_data: Three floating-point numbers for Vx, Vy, Vz velocity (currently placeholders).
6. checksum: A byte representing the checksum of the message. It is calculated by summing up all the bytes in the message before the checksum.

Example Message:  
<ST>;01;1696499957.774450967;-9.80,10.75,0.24;0.4,-0.3,-0.5,0.6;0.2,0.3,0.5;123;<EN>

## 

## **5. Configuration and Startup Routine**

### **5.1 Default Setup**

* **Baud Rate:** Default set to 115200.
* **Startup:** Initially, no messages are sent to the client; configuration is required.

### **5.2 Start Routine**

1. Power up.
2. Connect to the module.
3. Connect serially at 115200 baud rate.
4. Send Test Command and wait for acknowledgement.
5. Send SetDivisionRate command.
6. Receive position data.

### **5.3 Set Baud Rate Routine**

1. Power up.
2. Connect to the module.
3. Connect serially at 115200 baud rate.
4. Send Test Command and wait for acknowledgement.
5. Send ChangeBaudRate command.
6. Send Test Command and wait for acknowledgement.
7. Send SetDivisionRate command.
8. Receive position data.