Hose Jig System
Technical and User Manual

Generated by Cascade AI

July 25, 2025

Contents

1 Introduction

This document provides technical and user documentation for the Hose Jig system as implemented in the file `hose_jig.ino`. It includes wiring, hardware, command protocol, expected behaviors, and operational instructions.

2 Hardware Overview

2.1 Microcontroller and Interfaces

- **Platform:** ESP32
- **CAN Buses:**
  - CAN0 (TWAI, ESP32 Integrated)
  - CAN1 (MCP2515, External)
- **EEPROM:** For persistent counters and configuration
- **Servo Motors:** 3 (Left, Center, Right)
- **Linear Actuator:** X Axis
- **Inductive Sensors:** 3 (Left, Center, Right)

2.2 Pin Assignments and Wiring

Function	Pin	Details
CAN0 TX	GPIO4	ESP32 Integrated CAN TX
CAN0 RX	GPIO5	ESP32 Integrated CAN RX
CAN1 CS	GPIO16	MCP2515 Chip Select
CAN1 INT	GPIO17	MCP2515 Interrupt
Servo Left	GPIO12	PWM
Servo Center	GPIO13	PWM
Servo Right	GPIO14	PWM
Sensor Left	GPIO26	Inductive, INPUT extunderscore PULLUP
Sensor Center	GPIO27	Inductive, INPUT extunderscore PULLUP
Sensor Right	GPIO25	Inductive, INPUT extunderscore PULLUP

2.3 Other Hardware Details

- Servo PWM: 50Hz, 500-2500us pulse width
- Linear Actuator CAN ID: 0x2CE
- Device CAN ID: 0x0CA
- Response CAN ID: 0x4CA

3 Command Cases Overview

The system processes CAN instructions based on the first byte of incoming data. Below are all supported cases:

Case Code	Description
0x01	Reset microcontroller (restart)
0x02	Ping (read X axis status)
0x03	Home actuator (move to home position)
0x04	Move actuator to absolute position (angle, orientation)
0x05	Move all servos to open position
0x06	Move all servos to close position
0x08	Read servo movement counter
0x09	Reset servo movement counter
0x0A	Move all servos to specified position
0x0B	Move actuator to insertion position
0x0C	Move actuator to insertion position (duplicate)
0x0D	Read actuator movement counter
0x0E	Reset actuator movement counter
0x10	Update SERVO_OPEN_ANGLE
0x11	Update SERVO_CLOSE_ANGLE
0x12	Update ACTUATOR_DELIVER_POSITION
0x13	Update ACTUATOR_INSERTION_POSITION
0x14	Read SERVO_OPEN_ANGLE
0x15	Read SERVO_CLOSE_ANGLE
0x16	Read ACTUATOR_DELIVER_POSITION
0x17	Read ACTUATOR_INSERTION_POSITION
0xFF	Power off (move all to home position)
Other	Unknown command (returns error)

4 Command Input/Output Table

Case	Input Data (CAN)	Expected Output	Notes
0x01	[0x01, ...]	Device restarts, sends [0x01, 0x01, ...]	Reset MCU
0x02	[0x02, ...]	[0x02, 0x01, ...]	Ping reply
0x03	[0x03, ...]	[0x03, status, ...]	Home actuator, status: 0x01=OK, 0x02=Timeout, 0x04=No local network

0x04	[0x04, posH, posL, orient, ...]	[0x04, status, ...]	Move actuator to position
0x05	[0x05, ...]	[0x05, 0x01, ...]	Servos open
0x06	[0x06, ...]	[0x06, 0x01, ...]	Servos close
0x08	[0x08, ...]	[0x08, 0x01, counterH, counterL, ...]	Servo counter read
0x09	[0x09, ...]	[0x09, 0x01, ...]	Servo counter reset
0x0A	[0x0A, pos, ...]	[0x0A, 0x01, ...]	Servos to position
0x0B	[0x0B, ...]	[0x0B, status, ...]	Actuator to insertion position
0x0C	[0x0C, ...]	[0x0C, status, ...]	Actuator to insertion position
0x0D	[0x0D, ...]	[0x0D, 0x01, counterH, counterL, ...]	Actuator counter read
0x0E	[0x0E, ...]	[0x0E, 0x01, ...]	Actuator counter reset
0x10	[0x10, valH, valL, ...]	[0x10, 0x01, valH, valL, ...]	Update SERVO_OPEN_ANGLE
0x11	[0x11, valH, valL, ...]	[0x11, 0x01, valH, valL, ...]	Update SERVO_CLOSE_ANGLE
0x12	[0x12, valH, valL, ...]	[0x12, 0x01, valH, valL, ...]	Update ACTUATOR_DELIVER_POSITION
0x13	[0x13, valH, valL, ...]	[0x13, 0x01, valH, valL, ...]	Update ACTUATOR_INSERTION_POSITION
0x14	[0x14, ...]	[0x14, 0x01, high, low, ...]	Read SERVO_OPEN_ANGLE
0x15	[0x15, ...]	[0x15, 0x01, high, low, ...]	Read SERVO_CLOSE_ANGLE
0x16	[0x16, ...]	[0x16, 0x01, high, low, ...]	Read ACTUATOR_DELIVER_POSITION
0x17	[0x17, ...]	[0x17, 0x01, high, low, ...]	Read ACTUATOR_INSERTION_POSITION
0xFF	[0xFF, ...]	[0xFF, status, ...], then [0xFF, 0x01, ...]	Power off, move all to home
Other	Any	[0xFF, ...] (error)	Error response

5 Technical Manual

5.1 System Architecture

The Hose Jig system is built on an ESP32 microcontroller, utilizing FreeRTOS for multitasking and two CAN buses for communication: one integrated (TWAI) and one external (MCP2515). The system controls three servos and a linear actuator, with inductive sensors for feedback. EEPROM is used for storing counters and configuration.

5.2 Communication Protocol

- CAN0 (TWAI) is used for receiving main instructions and sending responses.

- CAN1 (MCP2515) is used for actuator control.
- Each command is a CAN frame with the first data byte as the case code.
- Replies use a separate response CAN ID (0x4CA).

5.3 Persistent Storage

Counters for servo and actuator movements, as well as configuration values (angles, positions), are stored in EEPROM to retain values across power cycles.

5.4 Error Handling

- Status codes: 0x01 (OK), 0x02 (Timeout), 0x04 (No local network)
- Unknown commands return an error response.

5.5 Troubleshooting

- Ensure all wiring matches the pinout table.
- Check CAN bus connections and termination.
- If actuators/servos do not move, verify power and CAN status.
- Use ping (0x02) to check device responsiveness.

6 User Manual

6.1 Setup

1. Connect all hardware as per the wiring table.
2. Ensure CAN bus termination resistors are present.
3. Power the ESP32 and peripherals.

6.2 Operation

1. Send commands over CAN0 (TWAI) using the correct device CAN ID (0x0CA).
2. Monitor responses on CAN0 with response CAN ID (0x4CA).
3. Use provided cases to control servos and actuator, and to read or reset counters.
4. For safe shutdown, use the power-off command (0xFF).

6.3 Maintenance

- Periodically check and reset movement counters as needed.
- Inspect wiring and connectors for wear or damage.
- Store configuration changes using the update commands (0x10–0x13).

6.4 Safety

- Always power off before servicing hardware.
- Avoid operating actuators or servos with obstructions present.
- Ensure CAN bus and power wiring are secure.

7 Appendix

7.1 Default Configuration Values

- SERVO_OPEN_ANGLE: 180
- SERVO_CLOSE_ANGLE: 0
- ACTUATOR_DELIVER_POSITION: 100
- ACTUATOR_INSERTION_POSITION: 100