Hose Puller Controller

Technical Documentation

Alan Silva
asilva@gswiring.com

July 25, 2025

Executive Summary

This document provides comprehensive technical documentation for the Hose Puller Controller firmware, which manages a robotic system for hose manipulation using an ESP32 with dual CAN bus support.

Contents

1 Overview

The Hose Puller Controller is an advanced robotic control system designed for automated hose manipulation. The system is based on an ESP32 microcontroller and utilizes dual CAN bus communication for controlling linear actuators, stepper motor, and digital gripper system.

1.1 Key Features

- Dual CAN bus communication (integrated TWAI + MCP2515)
- Linear actuator control for Y and Z axes
- High-precision stepper motor control
- Digital gripper system with force control
- Persistent counter storage in EEPROM
- Robust CAN command protocol

2 Hardware Configuration

2.1 ESP32 Pin Configuration

CAN Bus 1 - TWAI (ESP32 Integrated)

- **TX (Transmit):** GPIO4 (CAN0_TX)
- **RX (Receive):** GPIO5 (CAN0_RX)

CAN Bus 2 - MCP2515

- **CS (Chip Select):** GPIO16
- **INT (Interrupt):** GPIO17
- **SPI:** Uses default ESP32 SPI pins

2.2 Stepper Motor

Parameter	Value/GPIO
STEP Pin	GPIO25
DIR Pin	GPIO26
ENABLE Pin	GPIO27 (active LOW)
Steps per revolution	200 steps/rev
Maximum speed	1000 steps/s
Acceleration	500 steps/s^2

Table 1: Stepper Motor Configuration

2.3 Digital Gripper System

- **PWM Pin:** GPIO32 - Gripper force control
- **Open Pin:** GPIO33 - Open command

- **Close Pin:** GPIO34 - Close command

2.4 Linear Actuators

Linear actuators are controlled via CAN bus with the following addresses:

Axis	CAN ID
Y-Axis	0x2CB
Z-Axis	0x2CC

Table 2: Linear Actuator CAN Addresses

3 CAN Communication Protocol

3.1 Device Addressing

- **Device CAN ID:** 0x192
- **Response CAN ID:** 0x592

3.2 Command Reference

Command	Description	Data Bytes	Response
0x01	Reset microcontroller	-	-
0x02	Heartbeat	-	Status (0x01)
0x03	Home actuators (Y and Z)	-	Status (0x01=OK, 0x02=FAIL)
0x04	Move Y actuator	4 bytes (int32_t)	Status (0x01=OK, 0x02=FAIL)
0x05	Move Z actuator	4 bytes (int32_t)	Status (0x01=OK, 0x02=FAIL)
0x06	Read Z actuator counter	-	4 bytes (uint32_t)
0x07	Read Y actuator counter	-	4 bytes (uint32_t)
0x08	Reset Y counter	-	Status (0x01)
0x09	Reset Z counter	-	Status (0x01)
0x0A	Stepper motor (reserved)	-	-
0x0B	Move stepper	4 bytes (int32_t)	Status (0x01=OK)
0x0C	Home stepper	-	Status (0x01=OK)
0x0D	Open gripper	-	Status (0x01=OK)
0x0E	Close gripper	-	Status (0x01=OK)
0x0F	Set gripper force	1 byte (0-255)	Status (0x01=OK)
0xFF	Power off, home all axes	-	-

Table 3: Complete CAN Command Reference

4 EEPROM Memory Management

EEPROM Configuration

- **Total size:** 8 bytes
- **Address 0-3:** Y-axis movement counter (uint32_t)
- **Address 4-7:** Z-axis movement counter (uint32_t)

Counters are stored persistently to maintain actuator position tracking between system restarts.

5 System Dependencies

5.1 Required Libraries

Library	Purpose
ESP32-TWAI-CAN	Integrated CAN bus communication
mcp_can	MCP2515 controller for second CAN bus
SPI	SPI communication for MCP2515
ESP32Servo	Gripper system control
AccelStepper	Advanced stepper motor control
FreeRTOS	Real-time operating system
EEPROM (ESP32)	Persistent storage
Custom Libraries:	
src/linear_actuator.h	Linear actuator control
src/gripper_digital.h	Digital gripper system control

Table 4: System Dependencies

6 Status Codes

Code	Description
0x01	**OK** - Operation completed successfully
0x02	**FAIL** - Operation failed
0x03	**TIMEOUT** - Operation timed out
0x04	**NO LOCAL NETWORK** - Local network unavailable

Table 5: System Status Codes

Technical Support

For technical support or to report issues, contact Alan Silva at `asilva@gswiring.com`