

STM32 Homing Firmware

Test Task Documentation

Mohamed Aziz Ben Souissi

September 28, 2025

1 Overview

This project implements a **homing procedure** for an actuator using an STM32F407 Discovery board. The goal is to move the actuator between two limit switches (left and right), measure travel times, and compute the middle position.

The implementation is **modular and portable**, with clear separation between the homing algorithm and hardware access. A simulation backend allows running the code without physical hardware.

2 Project Structure

- **homing.c / homing.h** Core homing state machine (state transitions, debouncing, timing, error handling).
- **homing_porting.h** Hardware abstraction layer (HAL) defining the expected hardware functions:

```
1 bool HW_LeftSwitchRaw(void);
2 bool HW_RightSwitchRaw(void);
3 void HW_ActuatorMoveLeft(void);
4 void HW_ActuatorMoveRight(void);
5 void HW_ActuatorStop(void);
6 uint32_t HW_GetTickMs(void);
7 void HW_SimulationUpdate(uint32_t now_ms);
```

- **hw_simulation.c** Simulated actuator and limit switches. Enables testing on STM32 board without real motors/sensors.
- **homing_real_hardware.c** Hardware bindings for real actuator and GPIO-based switches.
- **main.c** Application entry point: initializes peripherals, starts homing after 3s, runs the state machine, and updates LEDs.

3 Homing Algorithm

The homing sequence follows these steps:

1. Move actuator to the **left limit** until the switch is pressed.
2. Move to the **right limit** and record travel time.
3. Move back **left** and record return time.

4. Calculate the **middle position** using measured times:

$$T_{middle} = \frac{T_{LR} \cdot T_{RL}}{T_{LR} + T_{RL}}$$

5. Move actuator to the middle and stop.
6. Set state = DONE (success) or ERROR (on timeout/fault).

3.1 Error Codes

- 1: Timeout while moving left
- 2: Timeout while moving right
- 3: Invalid middle calculation
- 4: Timeout returning left
- 5: Switch pressed unexpectedly during middle move

4 Simulation Behavior

The simulation backend models the actuator with fixed travel times:

- Left switch after 2s
- Right switch after 2s
- Return to left also 2s

This allows verifying homing cycles using only the STM32 LEDs.

5 LED Feedback

On the STM32F407 Discovery board:

- Orange (LD3): moving left
- Blue (LD6): moving right
- Green (LD4): homing done successfully
- Red (LD5): error condition

The cycle restarts automatically after a few seconds.

6 Testing Instructions

1. Flash the firmware onto the STM32F407 Discovery board.
2. Observe the LED sequence: orange \rightarrow blue \rightarrow blue \rightarrow green.
3. In case of timeout or switch fault, the red LED lights up.
4. The test auto-restarts for repeated demonstration.

7 Conclusion

The project successfully implements a homing procedure on the STM32 platform. Key achievements include:

- A clear separation between hardware-independent logic and hardware-specific drivers.
- A robust state machine that handles the full homing cycle (left \rightarrow right \rightarrow left \rightarrow middle).
- A simulation layer that allows validation without physical hardware.
- Integrated error handling with distinct error codes for different failure modes.
- Real-time visual feedback via on-board LEDs for each state.