

 AllCore360 / allcore-agent-balena-block- 



 Code  Issues  Pull requests 2  Actions  Projects  Security  Insights  Settings

Created mid-level abstraction #8

[Edit](#) [Code](#) [Jump to bottom](#)[Open](#) arthurtyukayev wants to merge 2 commits into [master](#) from [at/mid-level-hal](#) 

Conversation 24

Commits 2

Checks 0

Files changed 20



arthurtyukayev commented 2 weeks ago

This pull request introduces significant enhancements to the codebase, focusing on the addition of relay output (RO) hardware abstraction layer (HAL) support, the implementation of utility modules for system features, improvements to configuration handling, and the introduction of a robust unit testing framework with mocks. The Makefile is updated to incorporate these changes, enabling easier testing and modular development.

Major Additions and Improvements:

1. Relay Output (RO) HAL Implementation

- Added new hardware abstraction for relay outputs, including source (`ro.c`), header (`ro.h`), and address definitions (`ro_addr.h`). This provides a clean interface for relay channel control and encapsulates hardware-specific details. [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#)

2. Utility Modules for System Features

- Introduced utility modules for estop, rotation, and tilt functionality, each with clear interfaces and implementations to interact with hardware and relays. These utilities provide higher-level abstractions for reading sensors and controlling actuators. [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#) [\[6\]](#) [\[7\]](#) [\[8\]](#)

3. Configuration Handling Enhancements

- Updated `main.c` to demonstrate reading a configuration file, parsing its contents, and displaying both raw and parsed configuration data, improving visibility and debugging.

4. Makefile and Build System Improvements

- Modified the Makefile to:
 - Include new source directories for utilities and config.
 - Add targets for building and running unit tests with mock support.
 - Set up include paths for unit tests, ensuring compatibility and isolation from hardware dependencies. [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#)

5. Unit Testing Infrastructure with Mocks

- Added a stub for `linux/types.h` to enable building on non-Linux systems.

- Introduced a mock implementation and header for `piControlIf` functions, allowing tests to simulate hardware interactions and verify behavior.
- These changes enable robust unit testing of hardware-dependent code in isolation. [\[1\]](#) [\[2\]](#) [\[3\]](#)

Other:

- Updated `.ignore` to ensure the `plans` directory is not ignored by version control.



arthurtyukayev added 2 commits [2 weeks ago](#)

[Created mi-level abstraction](#) [8a75ccb](#)

[Updated testing and fixed issues](#) [99af137](#)

Copilot review requested due to automatic review settings [2 weeks ago](#)

Copilot [started reviewing](#) on behalf of **arthurtyukayev** [2 weeks ago](#)

[View session](#)

Copilot reviewed [2 weeks ago](#)

[View reviewed changes](#)

Copilot left a comment

Pull request overview

This pull request introduces a mid-level hardware abstraction layer (HAL) and utility functions for controlling a RevPi MIO system. It adds relay output (RO) support, utility modules for estop, rotation, and tilt functionality, comprehensive unit testing infrastructure with mocks, and updates to the build system. However, the PR has critical issues that prevent compilation.

Changes:

- Added relay output HAL with source files, headers, and address definitions
- Implemented utility modules (estop, rotation, tilt, helpers) for higher-level hardware control
- Created comprehensive unit testing framework with mock `piControl` interface
- Updated Makefile to support unit tests and new source directories

Reviewed changes

Copilot reviewed 19 out of 20 changed files in this pull request and generated 23 comments.

► [Show a summary per file](#)

💡 [Add Copilot custom instructions](#) for smarter, more guided reviews. [Learn how to get started.](#)



machine/src/utils/estop.c

Comment on lines +4 to +7

```
4 + int ReadEStop(void)
5 + {
6 +     return mio_get_di(4);
7 + }
```



Copilot AI 2 weeks ago

The ReadEStop function lacks documentation explaining what it does, what the return values mean (e.g., does 1 mean estop is active or inactive?), and why it reads from digital input channel 4. Adding a docstring would improve code maintainability.



[Copilot](#) uses AI. Check for mistakes.



Reply...

Resolve conversation

machine/src/utils/rotation.c

Comment on lines +6 to +10

```
6 + int ReadProxiRotate(void)
7 + {
8 +     return mio_get_di(1);
9 + }
10 +
```



Copilot AI 2 weeks ago

The function name "ReadProxiRotate" appears to be missing a letter or uses unclear abbreviation. It likely should be "ReadProximityRotate" or "ReadProxRotate" if "Prox" is a standard abbreviation in the codebase. The same issue exists with "ReadProxiTilt". Consistent, clear naming would improve code readability.

Suggested change

```
6 - int ReadProxiRotate(void)
7 - {
8 -     return mio_get_di(1);
9 - }
6 + int ReadProxRotate(void)
7 + {
8 +     return mio_get_di(1);
```

```
9 + }
10 +
11 + /* Backward-compatible wrapper; prefer ReadProxRotate in new code. */
12 + int ReadProxiRotate(void)
13 + {
14 +     return ReadProxRotate();
15 + }
```

Commit suggestion ▾

Add suggestion to batch

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

machine/src/utils/tilt.c

```
7 + {
8 +     return mio_get_di(2);
9 + }
10 +
```

**Copilot** AI 2 weeks ago

The function name "ReadProxiTilt" appears to be missing a letter or uses unclear abbreviation. It likely should be "ReadProximityTilt" or "ReadProxTilt" if "Prox" is a standard abbreviation in the codebase. This inconsistency with "ReadProxiRotate" suggests a naming pattern issue that should be addressed across the codebase.

Suggested change

```
10 -
10 +
11 + int ReadProximityTilt(void)
12 + {
13 +     return ReadProxiTilt();
14 + }
```

Commit suggestion ▾

Add suggestion to batch

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

machine/src/hal/ro.c

Comment on lines +29 to +30

```
29 + if (ro_offset_for_channel(ch, &offset, &bit) < 0)
30 + return -1;
```

**Copilot** AI 2 weeks ago

The if statement on line 29-30 is missing braces around the return statement. While this is syntactically correct for single-statement blocks, it's a common best practice to always use braces with control flow statements to prevent potential bugs when code is modified later.

Suggested change

```
29 - if (ro_offset_for_channel(ch, &offset, &bit) < 0)
30 - return -1;
29 + if (ro_offset_for_channel(ch, &offset, &bit) < 0) {
30 + return -1;
31 + }
```

Commit suggestion ▾

Add suggestion to batch

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

machine/src/hal/ro.c

Comment on lines +42 to +43

```
42 + if (ro_offset_for_channel(ch, &offset, &bit) < 0)
43 + return -1;
```

**Copilot** AI 2 weeks ago

The if statement on line 42-43 is missing braces around the return statement. While this is syntactically correct for single-statement blocks, it's a common best practice to always use braces with control flow statements to prevent potential bugs when code is modified later.

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

13 hidden conversations

[Load more...](#)

machine/test/unit/linux/types.h

Comment on lines +20 to +23

```
20 + /* ioctl macros - simplified stubs for testing */
21 + #define _IO(type, nr)      ((type) << 8 | (nr))
22 + #define _IOW(type, nr, sz) ((type) << 8 | (nr))
23 +
```

**Copilot** AI 2 weeks ago

The `_IOW` macro definition is simplified and ignores the `sz` parameter, which could mask bugs if tests depend on the actual `ioctl` number encoding. The comment says "simplified stubs for testing" which suggests this is intentional, but consider whether this could hide issues. If the actual `piControl.h` uses these macros and the test needs accurate values, this could be problematic.

Suggested change

```
20 - /* ioctl macros - simplified stubs for testing */
21 - #define _IO(type, nr)      ((type) << 8 | (nr))
22 - #define _IOW(type, nr, sz) ((type) << 8 | (nr))
20 + /* ioctl macros - more accurate stubs for testing, mirroring Linux encoding */
21 + #define _IOC_NRBITS      8
22 + #define _IOC_TYPEBITS    8
23 + #define _IOC_SIZEBITS    14
24 + #define _IOC_DIRBITS     2
25 +
```

View all 23 lines

Commit suggestion ▾

Add suggestion to batch

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

machine/src/main.c

```
45 +
46 + int main(void)
47 + {
48 +     ConfigMap map;
```

**Copilot** AI 2 weeks ago

The `ConfigMap` variable is declared but the type definition is missing since `config_loader.h` is not included in this PR. This will cause a compilation error. Ensure that `config_loader.h` and its implementation are added to the PR.

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

machine/src/main.c

```
60 +  
61 +     printf("=== Parsed config entries (%zu) ===\n", map.count);  
62 +     for (size_t i = 0; i < map.count; i++) {  
63 +         const ConfigEntry *entry = &map.entries[i];
```

**Copilot** AI 2 weeks ago

The ConfigEntry type is used but not defined since config_loader.h is missing. This will cause a compilation error.

[Copilot](#) uses AI. Check for mistakes.

Reply...

Resolve conversation

machine/src/main.c


Comment on lines +65 to +79

```
65 +         if (entry->direction == CONFIG_DIR_OUTPUT) {  
66 +             direction = "output";  
67 +         } else if (entry->direction == CONFIG_DIR_MEMORY) {  
68 +             direction = "memory";  
69 +         }  
70 +  
71 +         printf("%s.%s [%s] offset=%u bit=%u len=%u\n",  
72 +             entry->device, entry->name, direction,  
73 +             entry->offset, entry->bit, entry->length);  
74 +     }  
75 +  
76 +     printf("=== Input offsets ===\n");  
77 +     for (size_t i = 0; i < map.count; i++) {  
78 +         const ConfigEntry *entry = &map.entries[i];  
79 +         if (entry->direction != CONFIG_DIR_INPUT) {
```

**Copilot** AI 2 weeks ago

The CONFIG_DIR_OUTPUT, CONFIG_DIR_MEMORY, and CONFIG_DIR_INPUT constants are used but not defined since config_loader.h is missing. This will cause a compilation error.

[Copilot](#) uses AI. Check for mistakes.



Reply...

Resolve conversation

.ignore

Show resolved

Reviewers

 Copilot

Still in progress? [Convert to draft](#)

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

1 participant

