

AUSTRALIS

Introduction

This document serves to provide a comprehensive overview of the Australis Firmware system architecture and its implementation.

Related Documents

Available from <https://github.com/RMIT-Competition-Rocketry/>:

- AV2 Hardware Reference

Acknowledgements

Thank you to Aurora & Legacy project team leads Patrick Underwood and Brodie Alexander for providing the opportunity and environment to work on these rockets as part of the team, and thank you to everyone who helped make them a reality!

Key Contributors

Thank you to these contributors, without whom the Australis Firmware would not exist.

- Matthew Ricci – *Principal firmware developer*
- William Houlahan – *Initial driver implementations*
- Benjamin Wilshire – *Initial driver implementations*

Special Thanks

Other members of the Aurora V Avionics team:

- Hugo Begg – *avionics hardware*
- Jonathan Chandler – *ground control station*
- Jeremy Timotius – *data analysis*
- Lucas Webb – *ground control station*

Australis Documentation

Firmware Design Document

2025 AUSTRALIS AVIONICS

Version	1.0
Last Updated	2025/03/16
Date Created	2025/03/11

Contents

Important Terms 4

Abbreviations 4

1 **System Overview** 5

 1.1 Firmware Architecture 5

 1.2 Components 5

 1.2.1 Core 5

 1.2.2 Extra 5

2 **Implementation Details** 5

3 **Future Progress** 5

4 **Document History** 7

Appendix 8

 A Appendix Item 8

Important Terms

Australis	Student developed flight computer hardware platform for high power rockets.
Australis Core	An internal component providing the base API and critical logic; stylised <code>core</code> .
Australis Extra	An internal component providing modular systems that may be optionally included to extend core functionality; stylised <code>extra</code> .
Australis Firmware	Flight computer firmware system for high power rockets; designed for, but not limited to, deployment on Australis targets.
Component	A collection of semantically related code groups and files within the Australis Firmware ecosystem.
Device	A hardware element external to the controller that provides additional functionality via a connected interface.
Driver	Software implementation of a device or peripheral interface.
Peripheral	A hardware element internal to the controller that provides important extensions to the feature set of its core processor.
Submodule	An isolated system packaged within <code>extra</code> that extends system functionality to target source code. Submodules may only depend on the <code>core</code> API.
Target	A hardware platform on which the Australis Firmware operates.

Abbreviations

A3¹	Aurora 3
API	Application Programming Interface
AV2	Australis Version 2

¹Also refers to version 1 of the Australis flight computer hardware.

1 System Overview

1.1 Firmware Architecture

Australis Firmware is a software platform for implementing FreeRTOS task systems designed for high power rockets based on STM32 hardware. It is packaged as two internal components, `core` and `extra`, which can be integrated by developers to create a complete flight computer system for their required target.

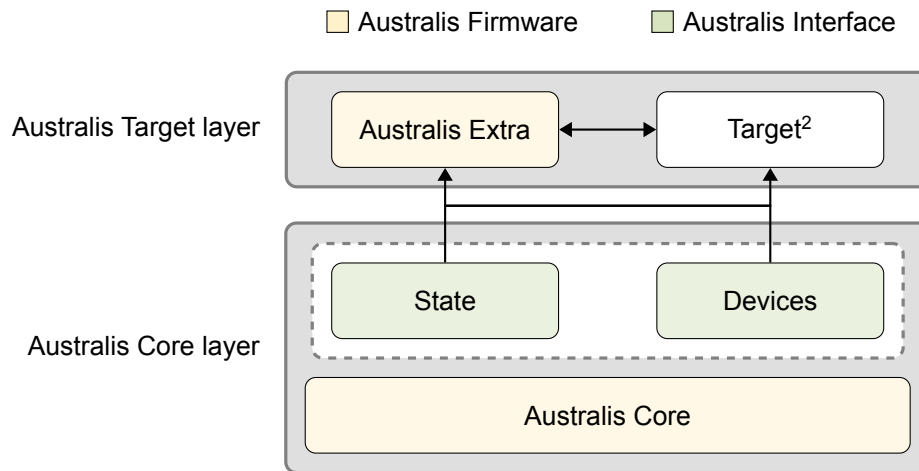


Figure 1.1: Australis Firmware system architecture

Australis Firmware implements a layered architecture (pictured Fig. 1.1), where `core` defines the base which provides the system API and critical logic, with the target layer on top integrating both `extra` submodules and target source code.

1.2 Components

1.2.1 Core

State

Devices

1.2.2 Extra

2 Implementation Details

3 Future Progress

²Target component is implementation specific.

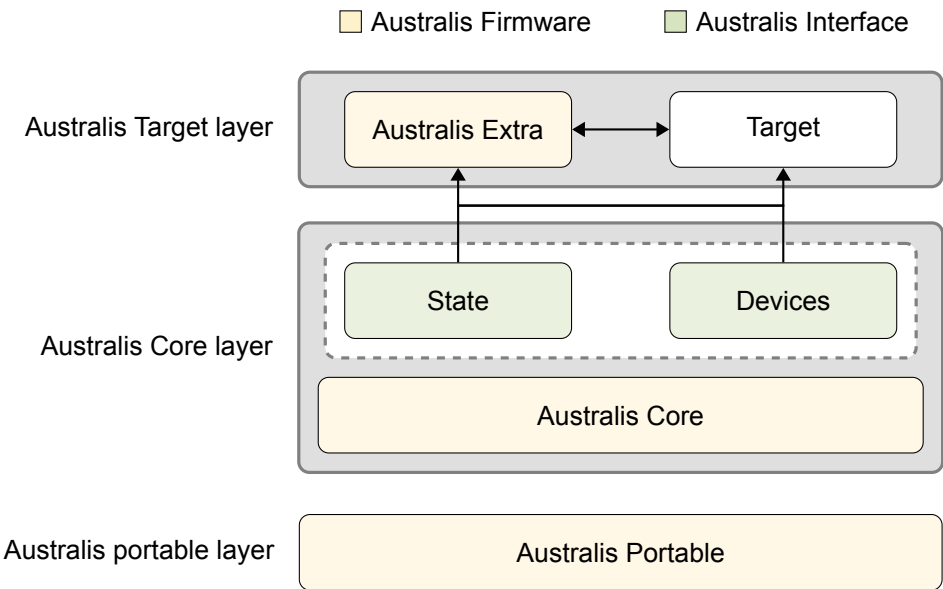


Figure 3.1: Proposed system architecture

4 Document History

Date	Changes Made	Made By
2025/03/16	Create initial document	Matthew Ricci

Appendix

Appendix A: Appendix Item