



Christiane Woletz

# **Firmware Development for a Distributed Sensor and Actuator Control System Using CAN Bus**

## **BACHELOR'S THESIS**

to achieve the university degree of  
Bachelor of Science

Bachelor's degree program:  
Software Engineering and Management

submitted to

**Graz University of Technology**

## **Supervisor**

Dipl.-Ing. Dr.techn. Tobias Scheipel, BSc  
Institute of Technical Informatics  
Embedded Architectures & Systems Group

Graz, August 2025



# Abstract

---

The increasing complexity of the hybrid rocket propulsion system for Aerospace Team Graz (ASTG)'s rocket *Halcyon* required the development of a new sensor and actuator control system. The goal of this thesis was to design and implement firmware and a custom communication protocol for Controller Area Network (CAN)-based communication, with a focus on reliability and scalability.

The CANary system consists of CANary nodes, which serve as distributed controllers for sensors and actuators. Each node communicates over a shared CAN bus using a custom protocol and supports dynamic configuration at startup. The same firmware is designed to run on all nodes, simplifying system maintenance and reducing configuration errors. The system was specifically developed to support engine tests and flight campaigns of the *Halcyon* propulsion system. It was used during twelve major system tests, including the European Rocketry Challenge (EuRoC) 2023 flight, which demonstrate reliable performance under real-world conditions.

The results show that the CANary system meets the requirements for distributed sensor acquisition, actuator control, and real-time communication in student rocketry applications. The system provides a scalable and maintainable platform for future propulsion and avionics developments at ASTG.



# Kurzfassung

---

Die zunehmende Komplexität des Hybridraketenantriebs des Aerospace Team Graz (ASTG)-Projekts *Halcyon* machte die Entwicklung eines neuen Sensor- und Steuerungssystems erforderlich. Ziel dieser Arbeit war es, Firmware sowie ein eigenes Kommunikationsprotokoll für die Controller Area Network (CAN)-basierte Kommunikation zu entwerfen und zu implementieren, mit besonderem Fokus auf Zuverlässigkeit und Skalierbarkeit.

Das CANary-System besteht aus CANary-Boards, die als verteilte Steuerungen für Sensoren und Aktuatoren fungieren. Jeder CANary kommuniziert über einen gemeinsamen CAN-Bus mit dem eigens entwickelten Protokoll und unterstützt eine dynamische Konfiguration beim Systemstart. Die Firmware ist so ausgelegt, dass sie auf allen CANaries identisch eingesetzt werden kann, was die Wartung vereinfacht und Konfigurationsfehler reduziert. Das System wurde speziell entwickelt, um Triebwerktests und Raketenstarts von *Halcyon* zu unterstützen. Es wurde in zwölf Gesamtsystemtests eingesetzt, darunter der Flug bei der European Rocketry Challenge (EuRoC) 2023, und zeigte dabei zuverlässige Leistung unter realen Bedingungen.

Die Ergebnisse zeigen, dass das CANary-System die Anforderungen an Sensorerfassung, Aktuatorsteuerung und Echtzeitkommunikation im studentischen Raketenbaukontext erfüllt. Das System bietet eine skalierbare Plattform für zukünftige Entwicklungen im Bereich Antrieb und Avionik des ASTG.



# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Thesis Structure and Organization . . . . .	2
<b>2</b>	<b>Background and Related Work</b>	<b>3</b>
2.1	CAN Bus . . . . .	3
2.2	AAUSAT3 and the CubeSat Space Protocol . . . . .	4
<b>3</b>	<b>System Overview</b>	<b>5</b>
3.1	Aerospace Team Graz (ASTG) Background . . . . .	5
3.1.1	System Architecture . . . . .	5
3.1.2	Hardware Specifications . . . . .	6
3.2	System Requirements . . . . .	7
<b>4</b>	<b>Communication Protocol</b>	<b>9</b>
4.1	Communication Protocol Design . . . . .	9
4.2	Communication Sequences . . . . .	11
4.2.1	Configuration . . . . .	11
4.2.2	Actuation . . . . .	12
4.2.3	Sensor Data . . . . .	12
<b>5</b>	<b>Software Design and Implementation</b>	<b>13</b>
5.1	Software on the CANaries . . . . .	13
5.1.1	Core Architecture and Task Management . . . . .	13
5.1.2	Configuration . . . . .	15
5.1.3	Controller Area Network (CAN) communication . . . . .	16
5.1.4	Sensor Data Acquisition With External ADC . . . . .	16
5.1.5	Local Sensor Processing . . . . .	17
5.1.6	Actuator Control . . . . .	17
5.2	Software on the Flight Computer . . . . .	18
5.2.1	CAN Communication . . . . .	19
5.2.2	Timestamp Synchronization . . . . .	19
5.3	Communication Protocol Implementation . . . . .	20

<b>6</b>	<b>Testing and Evaluation</b>	<b>21</b>
6.1	Run-Time Test . . . . .	21
6.1.1	Testing Setup . . . . .	21
6.1.2	Results and Discussion . . . . .	21
6.2	Field Testing . . . . .	22
<b>7</b>	<b>Conclusion and Future Work</b>	<b>23</b>
7.1	Conclusion . . . . .	23
7.2	Future Work . . . . .	24
	<b>Bibliography</b>	<b>25</b>
	<b>List of Abbreviations</b>	<b>27</b>
	<b>List of Figures</b>	<b>29</b>
	<b>List of Tables</b>	<b>31</b>

# CHAPTER 1

## Introduction

---

### 1.1 Motivation

Aerospace Team Graz (ASTG) is a student rocketry team primarily consisting of students of the Graz University of Technology. Founded in 2019, the team now consists of over 80 students from different fields of study. A goal of ASTG is to compete in the European Rocketry Challenge (EuRoC), an annual rocket launch competition for university teams organized by the Portuguese Space Agency.

The rockets used for the competitions in 2021 and 2022 employed solid motors, which did not require internal actuation or sensors for their propulsion systems. However, the rocket for 2023, Halcyon [1], uses a hybrid engine, which introduces more complexity. A hybrid engine uses propellants in two different states. The oxidizer is liquid, and the fuel is solid, with an additional gas used for pressurization. The oxidizer and pressurizer must be remotely tanked, for which new actuators and sensors were needed. A completely new part of the system is the filling station, which is used for the tanking procedure.

The requirement for additional sensors and actuators exceeded the capability of the previous flight computer, leading to the development of new sensor and control nodes. These nodes can be placed in the relevant parts of the rocket and communicate via a CAN bus with the flight computer. It contains eight sensor channels that are configurable for either thermocouples, pressure, and strain sensors, one high-power digital output, a Pulse-width Modulation (PWM) output for servos, and a potentiometer input. The CAN bus simplifies the addition of new nodes, allowing the same hardware to be reused multiple times: twice in the rocket, six times in the filling station, and also later in the test stand.

This thesis describes the firmware development for these control nodes, the design and implementation of a custom communication protocol used on the CAN bus, as well as the integration of the communication into the existing flight computer Real-Time Operating System (RTOS).

In particular, the firmware should sample sensors, convert data to the correct units, perform

basic filtering, and send sensor data on the CAN bus. It must also receive commands from the flight computer, actuate servo valves and digital outputs, and read potentiometer feedback. The firmware should be the same for all CANary instances.

## 1.2 Thesis Structure and Organization

Following the introduction, **Chapter 2** provides an overview of the background and related work about CAN bus and communication protocols.

An overview of the requirements and the system, where these sensor nodes have to be integrated, is provided in **Chapter 3**.

The communication protocol, the message structure, and different communication scenarios are explained in **Chapter 4**.

The software design is described in **Chapter 5**, including the common library for the communication protocol, the software for the CANaries, and the software for the flight computer.

In **Chapter 6**, the testing process is explained, with a description of the testing setup and the presentation of results.

A summary of the thesis and a discussion of possible future improvements are provided in **Chapter 7**.