



Danmarks Tekniske Universitet

---

## PROJECT ASSIGNMENT

---

COURSE:  
02249: CHP

AUTHOR(S):

Salik Lennert Pedersen (s134416)

Federico Bergamin(s162924)

Anders H. Opstrup (s160148)

**DTU Compute**

Institut for Matematik og Computer Science

---

02249: CHP

Carsten Witt

07 November 2016

**Abstract**

This document gives an overview of how to implement the Cubesat Space Protocol on the eCos operating system, document relevant features of Cubesat Space Protocol. It documents the currently used satellite communication protocol used at DTU, and compares these on a variety of factors. Also, implementation details are given regarding how the operating system specific subsystems have been implemented, along with notes on use of Error Correcting Codes. Basic testing shows equal operation of Cubesat Space Protocol on Linux and eCos.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
----------	---------------------	----------

# 1 Introduction

In 2010 Cubesat Space Protocol was created at Aalborg University. The protocol was created to use with the CubeSat satellite form in general and not only for Aalborg University. Cubesat Space Protocol has support for mechanisms such as packet checksums build in, and amongst other supports encryption. To make the protocol as generic as possible, the protocol was created with support for the operating systems Linux and FreeRTOS (later Mac OS X and Windows). Cubesat Space Protocol is designed with concurrency in mind, using the thread libraries of the relevant operating systems, and as of now no single threaded version of the protocol exists.

A CubeSat is a small research satellite contained in a 10cm × 10cm × 10cm box using mostly off the shelf electronic parts.

At DTU eCos is used as operating system on satellites. eCos is a real time openSource operating system, mainly intended for embedded use. eCos is highly customizable and can thus be designed for at very low resource footprint, making its use on satellites among other things a viable solution. eCos is based on GNU tools and bases its nature around having a POSIX compatible API. One of the key features in this situation is its support for threads, which is needed for implementing Cubesat Space Protocol which this project does.

This report will firstly outline some needed background knowledge in section ???. Section ??? describes the outline of the design requirements. One of the original reasons for this project, was to have Error Correcting Codes for the satellite link, but the goal has moved towards documenting the current protocol used at DTU and comparing it to Cubesat Space Protocol. This is done in section ???. Implementation specific details can be found in section ??, while details on testing the implementation comes in section ???. Section ??? describes the changing goal of the project in more detail along with ideas for ECC, and section ??? concludes the project.

This document requires previous knowledge to the protocols TCP and UDP and also the OSI reference model. Basic ECC understanding is needed to understand section ???.