Objectives

Introduction of onboard data handling concepts and characteristics
What Will be Said

- Satellite Elements
- Characteristics
- Purpose
- Operations
- Logical Model
- Architecture
- Functions
- Ørsted onboard data handling
- Robustness
- Software Development
- Pitfalls
- Résumé
Purpose

**Logistics**
- Power distribution
- Commanding
- Time synchronisation
- Status reporting

**Communication**
- With ground
- On satellite

**Autonomy**
- Handle platform & payload without ground contact

**Anomaly Handling**
- Maintain mission objectives
- Prevent loss of satellite
Characteristics

**Limited Resources**
- Processing power
- Memory
- Bandwidth on busses

**Embedded Real-Time Software**
- Hard real-time requirements
- Numerous events & actions
- Boot-strap software & application software
- No operating system (bare platform)

**Hostile Environment**
- Fault prevention
- Fault tolerance
Rømer Data Handling Context

Redundant Data Bus

TEST

MONS DPU

FM CHU

MONS CHU

RWA 0

RWA 1

RWA 2

RWA 3

CDH

DEBUG

STR CHU1

STR CHU2

PCDU

Com

ANT1

ANT2

Subsystems
Exercise: The Black Box

Perceive a computer with a piece of embedded software.

It is a block box revealing only a limited amount of status information.

What status information shall be available to assess the integrity of:

1: Hardware
2: Software
Onboard Traffic Management Service Provider 1

Onboard Traffic Management Service Provider 2

Onboard Traffic Management Service Provider n

Service Request

Service Reports

Service Requests

Service Reports

Service Request

Service Reports

Logical Model: Abstraction
Logical Model: Implementation

Space Segment

- Local Subsystem
- Provided Services
- Application Process 1

Data Handling & Control System

- Application Process 2
- Provided Services

- Onboard Traffic Management
- Remote Subsystem
- Provided Services
- Application Process N

Telecommand

Telemetry

Ground Segment

- Control Centre 1
- Control Centre 2
- Control Centre N
Packet Telecommand Standard
- Protocol for uplink: Ground segment ➞ Space segment
- Stream of telecommand packets

Packet Telemetry Standard
- Protocol for downlink: Space segment ➞ Ground segment
- Stream of telemetry packets

Packet Utilisation Standard
- Application layer: Ground segment vs. Space segment
- Logical model for satellite operation
Telecommand/Telemetry Formats

Telecommand

- **Head**
  - Destination ID
  - Sequence Counter
  - Command Type
  - Data
- **Tail**
  - Checksum

Telemetry

- **Head**
  - Originator ID
  - Sequence Counter
  - Telemetry Type
  - Time Stamp
  - Data
- **Tail**
  - Checksum
Functions

- Telecommand verification
- Housekeeping data collection
- Event reporting
- Memory read/write
- Function activation/deactivation
- Time synchronisation
- Command time line
- Parameter monitoring
- Telemetry storage
Telecommand Verification

**Telecommand Verification Service**
- Success/failure of telecommand execution
- Error code
- Telecommand identification

**Acceptance/Completion**
- Telecommand Packet ID
- Packet Source Control
- Code
- Parameters

**Progress**
- Telecommand Packet ID
- Packet Source Control
- Step Number
- Code
- Parameters
Housekeeping Data Collection

**Housekeeping & Diagnostics Data Reporting**
- Periodic reading of parameter values
- Reporting of temporally coherent values
- Typical period: 60 seconds
Event Reporting

- Nominal events
- Anomalies/errors
Memory Read/Write

Memory Management
- Reading/writing/verification
- Software updates (patching)
- Debugging/diagnostics

**Telecommand**

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Data</th>
</tr>
</thead>
</table>

**Telemetry**

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Length</th>
</tr>
</thead>
</table>

**Load**

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Data</th>
</tr>
</thead>
</table>

**Dump**

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Length</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Data</th>
</tr>
</thead>
</table>

**Checksum**

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Length</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Memory ID</th>
<th>N</th>
<th>Start Address</th>
<th>Length</th>
<th>Checksum</th>
</tr>
</thead>
</table>
Function Activation/Deactivation

Function Management
- Activation/deactivation of functions, modes etc.
- Execution of activities
- Functions identified by ASCII-string
Time Synchronisation

**Time Reporting**
- Generate time reports containing time stamp
- Report related to event on downlink
- Period based on required accuracy
Command Time Line

Onboard Scheduling
- Time line of telecommands
- Executed when due
- Updated based on operational schedule/time line
Parameter Monitoring

**Onboard Monitoring**
- Periodic reading of parameter values
- Comparison against nominal range
- Report deviations
- Initiate error handling

**Telemetry**

**Out-of-Limit report**

<table>
<thead>
<tr>
<th>N</th>
<th>Parameter#</th>
<th>Parameter Value</th>
<th>Limit Crossed</th>
<th>Transition Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
Telemetry Storage

Onboard Storage & Retrieval
- Storage of telemetry
- Inserted in downlink during ground contact
- Separate stores for different telemetry types
- Prioritised read-out
Exercise: Autonomy

Communication between satellite and control centre is possible 2 times 10 minutes per day.

The remaining time it must survive on its own.

What anomalies/event/situations should your satellite be able to handle autonomously?
Software Architecture
THE ØRSTED SATELLITE
Ørsted onboard data handling

**Hard Real-Time HOOD**

**Formal RAISE specifications**

**Automatic code generation + manual programming**

**Ada 83**

**Schedulability analysis: Deadline monotonic scheduling**

**In-Circuit emulator for software validation**

**Incremental development: Simulators ➔ prototypes ➔ final product**
Ariane 501

- Error in Inertial Reference System
- 64-bit float $\geq$ 16-bit integer = Overflow
- Chain of errors:
  - Reuse from Ariane 4; No revalidation
  - No exception handling
  - Post-mortem dump $\geq$ Valid input for data handling software
Robustness

Software Robustness Engineering

Fault Prevention
- Fault Analysis
- Supporting Methods & Tools
- Restricting Methods & Tools

Fault Removal
- Static Fault Removal
- Dynamic Fault Removal

Fault Tolerance
- Error Detection
- Error Recovery
Pitfalls

Interfaces: Focus from day one

Bidets: Estimate and survey

Schedulability: Estimate and survey

Fault tolerance: Keep it simple, stupid! (KISS)

Verification & validation: Unit test, integration test, system test ...
Résumé

*Embedded real-time software ➔ Tasking kernel*

*Real-time requirements ➔ Schedulability analysis*

*Fault tolerance ➔ Fail-stop*

*Numerous interfaces ➔ Interface control documents*

*Verification ➔ Simulators & prototypes*

*Keep it simple, stupid!*