*XMM Optical Monitor*

MULLARD SPACE SCIENCE LABORATORY
UNIVERSITY COLLEGE LONDON

Author: H.E.Huckle

# OM FM SOFTWARE TESTING AT MSSL

Document: XMM-OM/MSSL/SP/0207.2

**Distribution:**

| | | |
|---|---|---|
| XMM-OM Project Office | A Dibbens | Orig. |
| ESA PX | H Eggel | ✓ |
| CSL | P Rochus | |
| | S Roose | |
| Los Alamos National Laboratory | C Ho | ✓ |
| UCSB | T. Sasseen | ✓ |
| Royal Greenwich Observatory | R Bingham | |
| Mullard Space Science Laboratory University College London | R Card | |
| | M Carter | |
| | R Chaudery | ✓ |
| | J Fordham | |
| | H Huckle | ✓ |
| | R Hunt | |
| | H Kawakami | |
| | T Kennedy | ✓ |
| | D Self | |
| | P Smith | ✓ |
| | P Thomas | |
| | M Whillock | ✓ |
| | K Mason | |
| | A Smith | |

Author: Date:

OM Project Office Date:

Distributed: Date:

| Issue | Date | Comments |
|---|---|---|
| Draft | Dec '98 | Draft Version for Comment |
| | | |

# 1. Introduction

## *1.1 Document Overview*

This document describes the XMM-OM FM Software Testing in progress at Mullard Space Science Laboratory.

## *1.2 Acronyms and Abbreviations*

| | |
|---|---|
| BPE | Blue (Detector) Processing Electronics |
| CVS | Concurrent Version System |
| DBI | Data Bus Interface |
| DEM | Digital Electronics Module |
| DPU | Digital Processing Units |
| EGSE | Electrical Ground Support Equipment |
| ECR | Engineering Change Request |
| GNU | GNU Not Unix |
| ICU | Instrument Control Unit |
| RBI | Remote Bus Interface |
| NCR | Non Conformance Report |
| SSI | Serial Synchronous Interface |
| TMPSU | Telescope Module Power Supply |
| XMM-OM | X-ray Multi Mirror Mission Optical Monitor |

# 2. Software Testing Environment

## 2.1 Overview

The software test environment consists of 2 parts:-

- An emulation of the OM instrument consisting of
  - a DEM emulation based on the ICU and DPU breadboard models available to MSSL.
  - a software simulation of the telescope module on a PC running the Linux operating system.
- An emulation of the ground system and spacecraft based on a
  - Sun workstation running Solaris and used to
  - control a VME crate with the VxWorks real-time operating system running on a 68030 CPU card containing a card simulating the spacecraft interface.
- In addition, the Sun workstation is used to
  - cross compile and load the ICU ADA code.
  - load supplied DPU code.
  - version control the ICU code.

## 2.2 Hardware

The hardware consists of:

- A Sun workstation (including an archive disk).
- a VME crate containing
  - A Force30 68030-CPU based card running VxWorks.
  - The slot controller (SC) card in the VME rack, which is used to control the RBI interface to the ICU via the DBI.
- the ICU breadboard.
- the DPU breadboard.
- the SSI link between the ICU and DPU.
- A 300 MHz Pentium PC (running Linux) containing 2 interface cards
  - a MACSbus slave card to emulate the TMPSU and BPE MACSbus slaves on the FM.
  - an interface card to simulate the BPE 'data capture' interface from which detected events are forwarded to the DPU.
- Ethernet link between the Sun host and the 68030 CPU card in the VME rack.

## 2.3 Software

### 2.3.1 Solaris on the Sun Workstation

This provides the environment under which the VxWorks cross compiler and ADA cross compiler run. It also provides the standard Unix utilities for manipulation and examination of the archive datasets produced during a test session, such as `od` (dump), `grep` (text search) `diff` (difference) etc. In addition, various MSSL written utilities for the processing of the archive are based on this system, amongst which are:-

- Playback and display of the HK
- Image Display
- Telemetry Filters
- Comparison with standard datasets.

### 2.3.2 Tartan ADA

The Tartan ADA development system offers the following facilities:

- A validated ADA cross-compiler
  - 5 levels of optimisation
  - Can generate assembly listings
- Librarian
- Modular Runtime System
  - Storage Allocation, exception handling and tasking
  - Unneeded functionality can be excluded to reduce image size
  - Direct Connection of ADA procedures to interrupts
  - Replaceable priority-driven pre-emptive scheduler
- Linker with control of placement of code, data and constants for packages, modules, sections and subprograms.
- Assembler

### 2.3.3 Linux

This provides the software environment in which the emulation of the Telescope Module (OM1) is run. A PC was chosen due to the ease of adding interface cards. Linux was chosen in order to provide Unix level functionality (e.g. Multi-tasking, virtual memory, X-windows, networking, GNU C compiler etc.).

### 2.3.4 Telescope Module Simulator(s)

The primary C-based software simulator emulates the behaviour of the telescope module on receipt of the appropriate commands via the MACSbus for the following items:-

- Heaters.
- Dichroic (Beam Deflector).
- Filter Wheel (including the coarse and fine sensors).
- Flood LED.
- TPMSU Currents (between when secondaries are enabled or disabled).
- BPE High and Low Voltages (both Prime and Redundant).
- BPE Status Word.
- Acceptance of commands to load the centroid look-up table.
- Acceptance of commands to load the window look up table.
- Enabling and disabling of an integration.
- Setting of acquisition mode.
- Production of a stream of detector events consistent with i) a parameter specified image ii) spacecraft drift model and iii) the status of the simulated BPE (i.e. the integration enabled flag, window lookup table and acquisition mode). The inter-event time gap and photon confusion are correctly modelled. Note - part of this code is from the simulator used by American consortium.
- Correct reporting (for the purposes of HK) of the above.

There is a second C-based software simulator. Its purpose is to produce as high a count rate as possible in order to fully the stress the code. In order to a achieve this it has a lower functionality, concentrating only on the production of a stream of detector events consistent with a parameter specified image. The contents of the event stream is not determined by commands via the MACSbus but by a set of configuration files. The inter-event time gap and photon confusion are correctly modelled.

VxWorks

VxWorks is a real-time operating system with the following features:

- Multi-tasking support.
- Pre-emptive priority-based scheduling.
- Task synchronisation and resource exclusion facilities (‘semaphore’s).
- A text based command interface (the ‘C-shell’) with a Unix ‘flavour’ for consistency with the Sun Host
- The ability to run ‘scripts’ - a file containing a list of commands - as if they had been typed in at the main console. These are used to emulate various spacecraft operations scenarios.
- A close coupling with the file system of the host Sun for ease of data storage and recovery, in particular for the archiving of telecommands sent and telemetry received during a test.
- Facilities to develop tasks that can be initiated on VxWorks. These tasks can then, optionally, communicate directly with the host (using standard Unix ‘socket’ communications).
- Support for ‘logon’ from the host.

**Note**: The Sun host is used to cross-compile C code using a GNU C Motorola 68030 compiler. The code is loaded onto the card over the ethernet link using the standard TCP/IP protocol.

### 2.3.5 CVS

**cvs** is a version control system with the following features:

- extends the notion of revision control from a collection of files in a single directory to a hierarchical collection of directories consisting of revision controlled files.
- the directories and files can be combined together to form a software release.
- provides the functions necessary to manage these software releases
- controls the concurrent editing of source files among multiple software developers.
- keeps a single copy of the master sources called the source ``repository''
- permits extracting previous software releases at any time based on either a symbolic revision tag, or a date in the past.

# 3. Test Procedures

## 3.1 Overview

The MSSL test environment is used to control, modify and test ICU code and check out supplied DPU code from our American colleagues. This is done in a context that simulates, as far as possible, the whole OM instrument. It is also used to investigate interactions between the ICU and DPU codes.

Diagnostic information is obtained by a variety of techniques.

- Monitoring of the Housekeeping in the context of the commands sent.
- Monitoring of the Non-periodic housekeeping (issued by both the ICU and DPU) in the context of the commands sent. The ADA exception packets are particularly useful.
- Monitoring of 'debug' non-periodic packets which use the reserved SID range 20 -> 2F (hex).
- Monitoring of reserved areas of memory into which diagnostic information is written and which are then accessed by direct memory access from the host system. This avoids the latency and potential complexity of packet construction and sending involved in the above techniques.
- Examination, again in the context of the commands sent, of the displays of OM1 simulator which indicate its internal state.

## 3.2 Module Level

At this stage of the OM code development process, module level testing is generally not performed as the overall interface and internals of the modules have been well established since the engineering model. Should the need arise, an *ad-hoc* test harness, essentially consisting of a suitable main program to act as a driver of the module would be written. It is then compiled and linked with the module under test and the result loaded into the breadboard ICU. The code is then run and the result of the test noted.

## 3.3 System Level

For each NCR or ECR raised, the following procedure is adopted:

- The appropriate script or scripts that will exercise the problem are selected
- A review and selection of suitable additional diagnostic code takes place.
- If necessary, appropriate modifications to the Telescope Module simulator code are performed (e.g. additional diagnostics on its internal state could be added)
- The DPU and ICU codes are exercised using one or more of the standard VxWorks scripts which emulate various OM operational scenarios. An outline of these scripts and the commands from which they are constructed is given in XMM-OM/MSSL/ML/0006 , "XMM-OM VxWorks Command summary (FM MODEL)".
- The telecommands and telemetry are archived and examined using the tools described above.
- The problem is diagnosed and appropriate modifications are made to the code.
- The cycle is repeated until the problem is solved.
- The modified code is entered into the CVS repository
- The ECR or NCR is formally closed with a test report summarising the above.

# MSSL S/W Test Environment