Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2002-590

An independent verification tool for multi-vendor Mode S airborne transponder conformance testing

L.J. Aartman, P.J. van Heijningen, P. Brun and F. Ziegler

NLR-TP-2002-590

An independent verification tool for multi-vendor Mode S airborne transponder conformance testing

L.J. Aartman, P.J. van Heijningen, P. Brun* and F. Ziegler**

 *   *Eurocontrol Experimental Centre (France)*
**   *CIMS GmbH (Germany)*

Abstract

This paper describes the development of an automated laboratory test bench named MODEST (Mode S Transponder Test Tool). The objective of the tool is to perform vendor independent verification of Mode S transponders, with emphasis on basic functionality and protocols. A set of test procedures as defined in EUROCAE MOPS ED-73A for Mode S transponders have been translated into automatically executable test suites. The test tool is of a modular design, built around a Windows NT-based generic and re-usable test kernel ARTIS (Avionics Research Test and Integration System). ARTIS is extended with several application specific components for RF, ARINC-429, ARINC-718 data link interface and discrete I/O to interface with the transponder. The tool engineering process has been guided by the tool qualification approach defined in RTCA-DO178B, from Tool Operational Requirements to Tool Qualification. After a first delivery of the basic tool set early 2002, an upgrade of the tool with test suites to verify the ACAS-transponder protocols and interface in accordance with ED-73A and ARINC-735 is under development.

Abbreviations and Acronyms

ADLP	Airborne Data Link Processor
ACAS	Airborne Collision Avoidance System
ARTIS	Avionics Research Test and Integration System
ATM	Air Traffic Management
DLL	Dynamic Link Library
ECAC	European Civil Aviation Conference
EEC	Eurocontrol Experimental Centre
ICAO	International Civil Aviation Organization
MODEST	Mode S Transponder Test Tool
MOPS	Minimum Operational Performance Specification
MRPB	Most Recent Parameter Buffer
NLR	National Aerospace Laboratory NLR
PCO	Point of Control and Observation
RF	Radio Frequency
SI	Surveillance Identity (code)
TCAS	Traffic Collision Avoidance System
XPDR	Transponder

Contents

(22 pages in total)

Introduction

Background

Mode S is based on an enhancement of the Secondary Surveillance Radar (SSR) system and is currently deployed throughout the whole European Civil Aviation Conference (ECAC) area in accordance with the ECAC En-route Strategy for the 1990's. In addition to SSR functionality, Mode S enables the selective interrogation of aircraft, the extraction of air-derived data and the exchange of data through which new ATM functionality can be implemented. Furthermore, SSR Mode S facilitates the use of Airborne Collision Avoidance Systems (ACAS/TCAS). The Mode S standards are defined in ICAO Annex 10 Volumes III and IV [1][2]. The initial deployment of SSR Mode S Elementary and Enhanced Surveillance in Europe is planned for the high-density traffic area of the ECAC states, by both the civil and military administrations in this area. Mode S Sensors are also installed in large numbers in the U.S.A. The European civil administrations that have indicated that they are planning implementations so far include Belgium, Denmark, France, Germany, Luxembourg, Switzerland, The Netherlands and the United Kingdom. The present programme milestones [3] are:

£   Mode S Elementary Surveillance operational deployment from 2003, and

£   Mode S Enhanced Surveillance operational deployment from 2005.

Airborne Equipment

The prime on-board equipment required for Mode S surveillance is the Mode S transponder. It enables multi-functionality when interconnected with other airborne equipment (see fig. 1) [4]. Detailed definitions of all the parameters required for Elementary and Enhanced Surveillance are given in the ICAO Manual on Mode S Specific Services [5]. Transponder certification is further guided through the Minimum Operational Performance Specifications as laid down in RTCA-181C [6] and EUROCAE ED-73A [7]. These MOPS documents are intended to ensure that aircraft Mode S SSR transponders certified to it will be compatible with ICAO Annex 10 standards. They include performance characteristics and tests under normal operating conditions and simulated airborne conditions. ED-73A provides requirements and tests for several areas including Level 5 transponders, Extended Squitter Functions, ACAS Change 7, ACAS Crosslink and SI code capability [7].
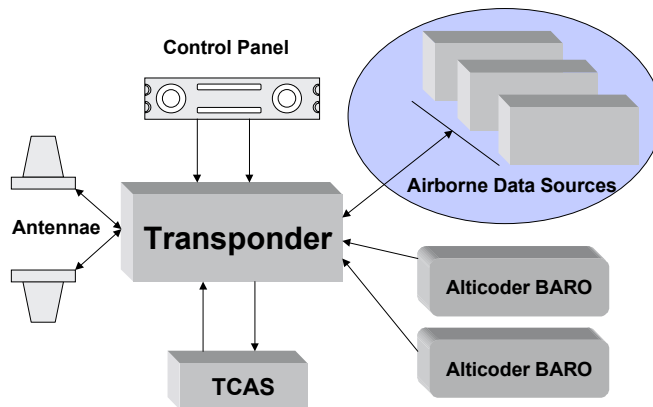
Figure 1 Airborne configuration

For the initial implementation of Mode S Elementary Surveillance, at least an ICAO Level 2 transponder system is needed to allow a ground interrogator to extract the required parameters from the aircraft [3]. The Mode S transponder replaces the Mode A/C transponder (used in former installations) on the aircraft. Reliable and safe operation of the Mode S communication requires proper functioning of the variety of Mode S transponders manufactured by multiple vendors, and an adequate aircraft installation. Despite all efforts from defining international aeronautical telecommunication standards, careful design assurance during transponder development and associated certification, it is anticipated that problems may arise with Mode S avionics equipment and aircraft installations in the initial stages of implementation.

This has resulted in the definition of the need for an independent validated test bench to collect additional objective evidence about the performance of transponder products from multiple vendors.

Monitoring will take place using a transponder test bench by the Eurocontrol Experimental Centre EEC and by fleet monitoring using State Facilities before and during the deployment of the introduction of Mode S Elementary and Enhanced Surveillance. Both problems observed during fleet monitoring or reported through Anomaly Reports may require further investigation. In case deviations from the specifications are detected, manufacturers (either original equipment manufacturers, tool manufacturers or airlines) will be notified immediately and delicately through confidential fault reports by the test operator.

Objective

The objective of the laboratory test bench is to perform a vendor independent verification of transponder compliance with a subset of the requirements of EUROCAE ED-73A. Verification is done through executing a predefined subset of test procedures defined in EUROCAE ED-73A.

Tool Operational Requirements

Although not mandatory, the tool engineering process that was conducted for the <u>Mode S</u>
Transponder <u>Test</u> Tool MODEST used the tool qualification approach defined by RTCA-
DO178B as a guideline [8]. DO-178B defines the requirement to perform tool qualification if a
tool is used to automate software development processes or software verification processes that
are typically performed by humans. Tool Operational Requirements (TOR) must be provided,
and the tool must be verified against the operational requirements.

Summary of Key Requirements

MODEST has to automatically execute selected test procedures from the ED-73A MOPS for
Mark 3 Mode S transponders. The selected tests will only be related to communication
protocols and not to signal levels and low-level timing.
The following test cases need to be supported by the first release of MODEST:

Table 1 List of implemented ED73A tests

Section	Test
5.4.12.2.2	Variable Direct Data
5.4.12.3	Standard Transaction Interfaces
5.4.12.4	ELM service interfaces
5.4.12.5	Comm-B Downlink Interface, Message Cancellation
5.4.12.6	Downlink ELM Interface, Message Cancellation
5.5.8.1	PROCEDURE #1 Error protection
5.5.8.2	PROCEDURE #2 Interrogation Acceptance Test
5.5.8.3	PROCEDURE #3 CA Verification
5.5.8.4	PROCEDURE #4 Non-Selective Lockout Tests
5.5.8.5	PROCEDURE #5 Selective Lockout Tests
5.5.8.6	PROCEDURE #6 Squitter verification
5.5.8.7	PROCEDURE #7 FS & VS Protocol/Code Tests
5.5.8.8	PROCEDURE #8 PI (Parity/Identity) Verification
5.5.8.9	PROCEDURE #9 Address tests
5.5.8.10	PROCEDURE #10 Altitude tests
5.5.8.11	PROCEDURE #11 4096 code test
5.5.8.12	PROCEDURE #12 RI, AQ Max Air
5.5.8.13	PROCEDURE #13 PR tests
5.5.8.15	PROCEDURE #15 Comm-A interface
5.5.8.16	PROCEDURE #16 Broadcast all calls

Section	Test
5.5.8.20	PROCEDURE #19 AIS Flight ID
5.5.8.21	PROCEDURE #20 Capability report
5.5.8.22	PROCEDURE #21 Directed Comm-B
5.5.8.23	PROCEDURE #21A Comm-B Broadcast
5.5.8.24	PROCEDURE #22 Downlink Interface, Storage Design, Buffer Rate
5.5.8.25	PROCEDURE #23 Downlink Interface, No-storage Design
5.5.8.26	PROCEDURE #24 Comm-C protocol
5.5.8.27	PROCEDURE #24A Enhanced Comm-C Protocol
5.5.8.28	PROCEDURE #25 Uplink Interface, ELM-Comm-C

Other key requirements for MODEST are:

monitor and control all transponder interfaces as required by the test procedure;

take the transponder level into account when reaching a verdict in tests that are transponder level and class dependent;

allow the modification of existing tests suite and test cases, and adding new test cases;

provide detailed test results (in terms of what was controlled, observed and expected);

support timing of transmission and receptions with a resolution of 1 ms.

execute all test cases for a given transponder within no more than two weeks;

MODEST has to run on a PC-platform using Windows NT[1]. Test cases have to be implemented in the C programming language. The tool has to be able to control an existing Mode S Data-Link Ground station (supplied by EEC) that provides the RF interrogation signals and accepts and decodes the reply signals on the RF interface with the transponder under test.

Avionics Research Test and Integration System (ARTIS)

MODEST uses NLR's Avionics Research Test and Integration System ARTIS, a re-configurable concept for testing avionics and used in a number of similar test applications at the NLR since 1996 [9]. An ARTIS test tool consists of a generic ARTIS kernel and specific components. An internal test bus is created depending on the configuration (fig. 2). The ARTIS kernel offers logging, monitoring and automatic testing functionality whilst test application dependent add-ons handle specific adaptations towards the Unit Under Test.

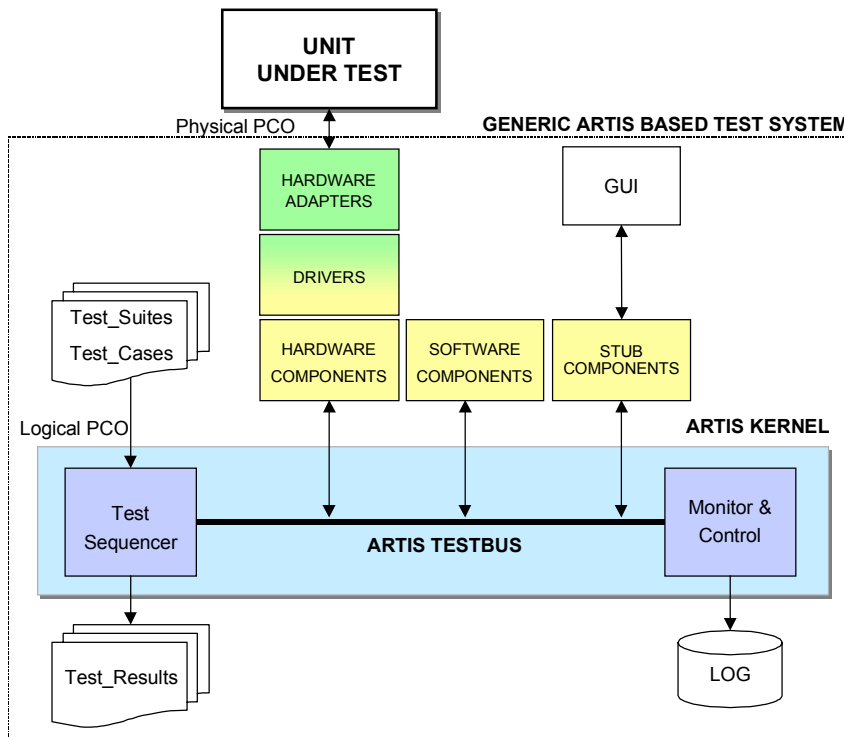[1] Windows NT is a registered trademark of Microsoft Corporation

Figure 2 Generic ARTIS-based test system

The components implement the functionality that is required for a specific Unit Under Test.
ARTIS uses three component types that are implemented as Dynamic Link Libraries (DLL):
Hardware components, to interface the ARTIS test bus with the actual hardware device
drivers;
Stub components, Graphical User Interface elements to allow interaction between the test
operator and the test bus, thus allowing to issue manual stimuli and direct observation
presentation;
Software components, to process parameters exchanged via the test bus. These software
components can for instance consist of an executable model, created with a functional
modeling tool [9].

ARTIS uses a 'publish and subscribe' mechanism to exchange parameters on the test bus,
containing the Most Recent Parameters. For automated testing the test cases interface with the
test bus through Points of Control and Observation (PCOs). The PCO is a concept used to
define the connection points or interfaces to the Unit or Interface Under Test for sending and
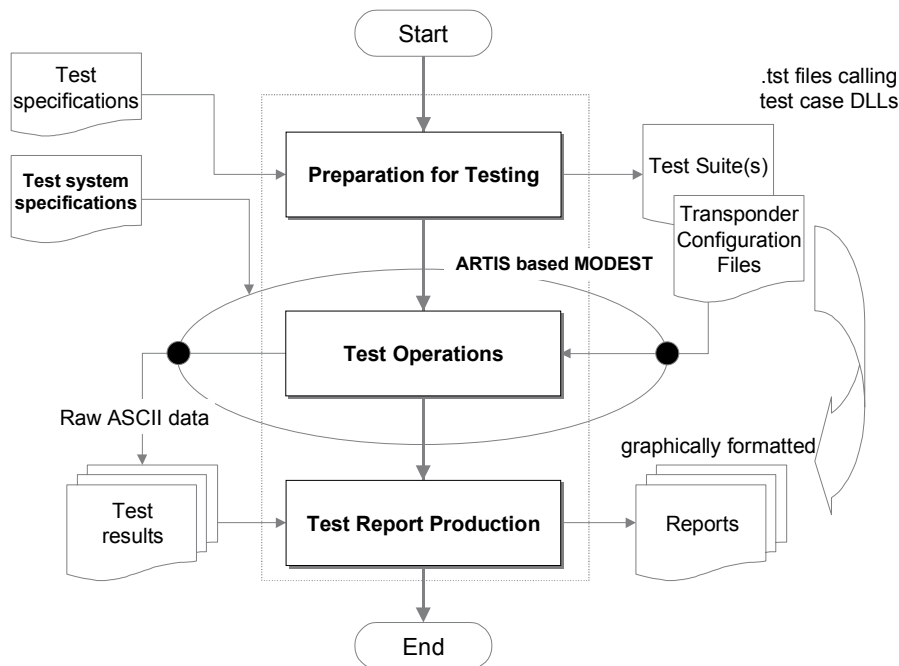receiving ARTIS messages [10].

Figure 3 From Specification to Test Report

Conformance Test Approach

The methodology of MODEST follows the three major steps for conformance testing (fig. 3):

[1]   Test Preparation, the translation of test procedures into automatically executable test cases, including the definition of the test configuration;

[2]   Test Execution, the actual testing, using the test cases or test suites as input, providing raw test results as an output;

[3]   Test Report Production, is the process of creating the necessary (formatted) test reports out of the raw test results.

Tool Development Environment

A number of small database applications (based on MS-Access) were used throughout the development of MODEST, to capture the properties of all internal and external interfaces. This concerns among others electrical signals, associated software parameters and the verdicts made by the test scripts, in order to achieve a high degree of integrity, and to support the traceability into formal tool qualification. The generated reports are not only a basis for the interface and design documentation, but are also used to generate common header files for the ARTIS components used to compose MODEST.

Translation of Test Procedures into Automatically Executable Test cases

A set of test procedures as defined in EUROCAE MOPS ED-73A for Mode S transponders have been translated into automatically executable test cases.

EUROCAE ED-73A uses plain text to describe which testing activities are to be executed and which results are to be expected. The plain text test procedures were by far not detailed enough to specify directly the correct stimulation of the transponder at the required detail from the test procedures only. It should be noted that a single wrong field value can stimulate the transponder into a state in which it does not react as expected in a particular test. It was therefore necessary to carefully reflect all requirements of Annex 10 to correctly specify all individual stimuli representing the plain text test requirements of ED-73A. Figure 9 presents an example of the text describing a test in ED-73A and the resulting code which implements this particular test section.

After the initial version of a test had been converted, the test procedure was executed on MODEST with a reference transponder for initial validation. The detailed test reports were analysed and discrepancies with Annex 10 and ED-73A were corrected. When proper functioning of the reference transponder was determined, the procedures were again validated against the MOPS and Annex 10 for correctness.

The test language used to implement the test procedures was ANSI C. This makes the test procedure implementation most comprehensive, flexible and allows all necessary automated code to be implemented.

A function 'send' is used to send a stimulus to a particular PCO. The stimulus data itself is stored in a data structure which is filled with the appropriate data prior to calling the send function. Together with the data also the time past test start (in ms) is passed as an option to define the moment when the stimulus has to be actually issued. This relieves the test case processing from hard real-time requirements.

If an observation is to be obtained, then a function 'receive' is called. This function fills an observation data structure with the observed data. Its parameters indicate on which interface (PCO) it should await data, where to store the received data and how long it should wait for the observation. The timeout prevents deadlocks in case the observation is not generated by the transponder as expected. When the receive function returns, it indicates the actual time, when the observation has been made by ARTIS. It also provides information whether data has been successfully received or a time-out has occurred.

The observation structure can then be analysed in a 'compare' function for correctness. This function verifies if the recorded observation structure content is correct, by comparing the observation with the expected field values. The return value of compare is either PASS or FAIL.

To be able to write the code in a clear and understandable manner, the data structures mentioned above are explicitly filled with individual field values without any bit manipulations.The resulting tests are compiled as DLLs that are loaded for execution by the test system.

Transponder Configuration Files

At run time each Transponder Under Test requires its own *Transponder Configuration File*. This file is prepared using a small database application before starting test operations. The file specifies the transponder/aircraft relationship, the transponder's capabilities and default environment values to be used during the test.

Test Report Production

While executing a test, MODEST generates raw test result files in ASCII format. These result files are comprehensive and contain timestamp, comments and data for each individual test event. It precisely reflects which stimuli were sent to the transponder under test and how the transponder reacted. Time stamping is done in milliseconds relative to the start of the ARTIS kernel. The raw result file can be converted into a graphically formatted Test Report using the MODEST Report Production database. Each result file is concluded with a verdict: 'Pass' or 'Fail' which provides an overall test passing criteria.

During report production the Transponder Configuration File can also be included in the report using integer data relationships. Figure 10 provides an example raw test result file. The following table explains the identifiers used.

Table 2 Result file identifiers

| Ident	Description	Purpose
CC	Comment Case	Generated by the test case
CS	Comment Suite	Generated by the suite
DC	Data Controlled	Provided stimuli
DE	Data Expected	Only reported when expected data is not equal to observed data
DO	Data Observed	Observation performed
Hn	Header [0..n]	Header information like date, time, Configuration Identification
TC	Test Case	Identifier of TEST case in suite
VC	Verdict Case	At the end of the test case
VS	Verdict Step	Result of the compare (PASS/FAIL)
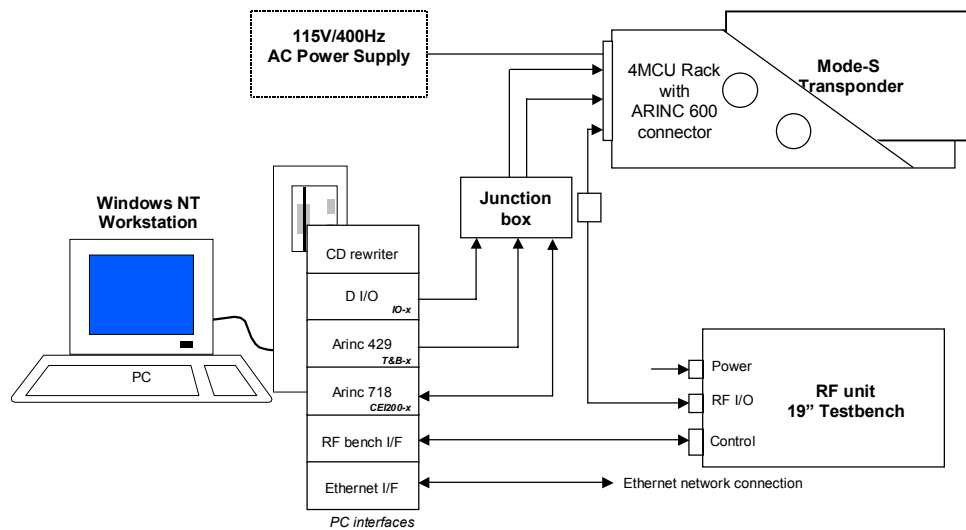
Figure 4 MODEST hardware architecture

MODEST Characteristics

Basic Architecture
The MODEST test tool is of a modular design built around a Windows NT-based generic and re-usable test kernel ARTIS. It runs on an industrial PC extended with several PC interface adapters for RF, ARINC-429, ARINC-718 data link interface and discrete I/O (fig. 4, 6).

MODEST Points of Control and Observation
The virtual interface between the test cases and the Transponder Under Test is defined by PCOs (fig. 5). This mechanism provided by ARTIS forms an abstraction layer for the test cases and relieves the test cases from the low level signal management. Signal management is being cared for at the test bus and the components. A PCO can be defined as a boolean, an integer or a complex structure like a Mode S Downlink packet across the ARINC 718 data link interface or the SSR Mode S format handled by the interrogator. From the test case perspective PCO drivers are in charge of ensuring proper conversion between logical and physical representation of the PCO values. In ARTIS, the PCO driver consists of a chain. Data is entered at the logical PCO as data structures. An initial conversion of the PCO data structures takes place in a conversion module `ed73a_mops.c`, which converts the data into a representation recognised by ARTIS. The data then progresses via the test bus, the individual components and the associated driver into the adapter card that physically interfaces the Transponder Under Test. For example the altitude PCO driver transforms the logical 'altitude' variable value to the physical stimulus (i.e. either Gilham discretes or air data computer ARINC words). Another PCO driver records the

signals on the ARINC 718 bus and derives the actual values of the related logical data structure fields from this signal or vice versa. To further enhance the test script processing and the maintainability, stimulus and observation data are represented by means of structures. Each component of the structure represents the value of a logical field of the individual elements of ARINC 718 records or RF signals. This allows dealing with individual parameter values of a PCO simply by manipulating a component of the related structure rather than accessing particular bits in a byte stream. An additional advantage of this abstract view is that the test script becomes independent from the actual physical representation (and also from the interface actually used by the transponder). This approach does however require that the PCO driver correctly performs the conversion between physical and logical representation. The list of required PCOs (table 3) has been defined for each ED-73A test procedure to be implemented during the requirements analysis phase.
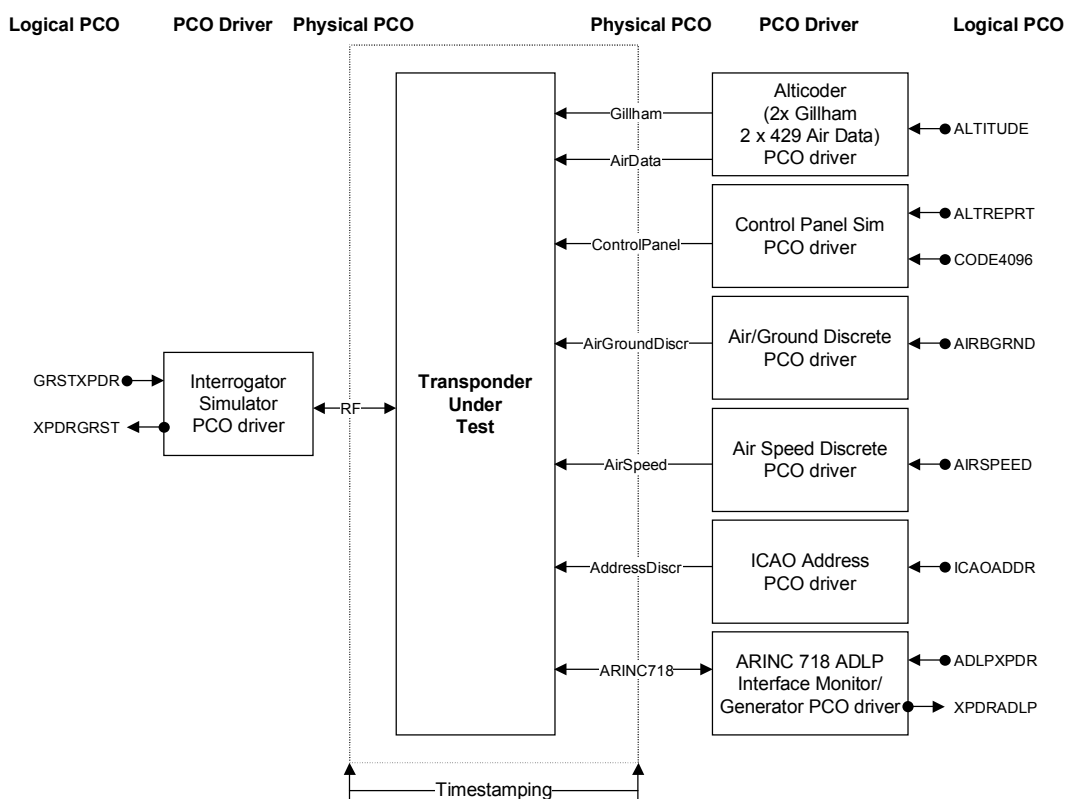

Figure 5 PCO overview from test case perspective

For MODEST the following PCOs have been defined:

Table 3 List of MODEST PCOs

| PCO	Description
ALTITUDE	Altitude input to the transponder
ALTREPRT	Altitude Report Enable
CODE4096	4096 Code
AIRBGRND	Air/Ground Discrete
AIRSPEED	Airspeed
ICAOADDR	24 bit ICAO aircraft address
ADLPXPDR	ADLP to transponder transfers
XPDRADLP	Transponder to ADLP transfers
GRSTXPDR	Interrogator Uplink transmission
XPDRGRST	Interrogator Down link transmission
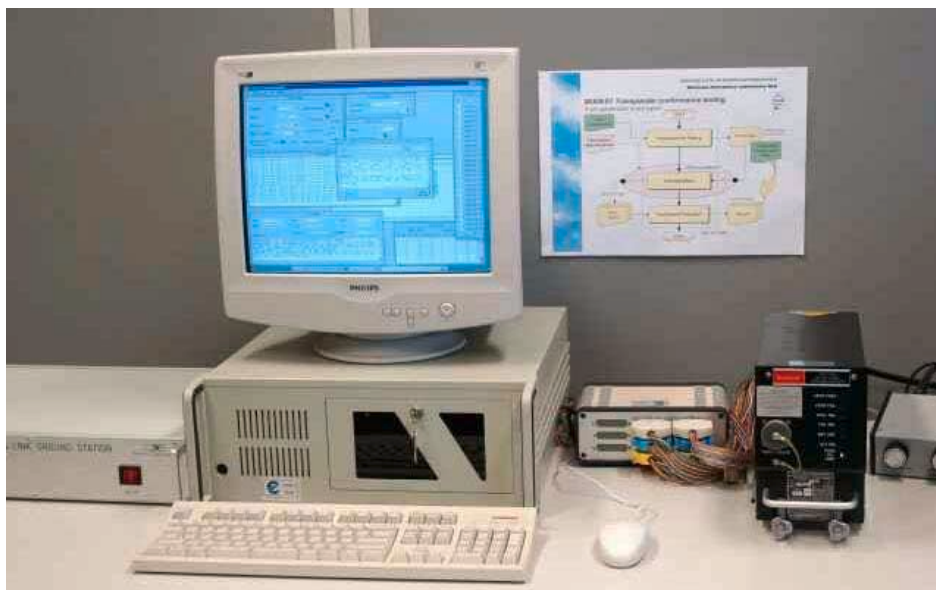
Figure 6 MODEST setup

Major Components

A *hardware component* describes how to exchange the detailed messages between an external device connected via a hardware interface and the Most Recent Parameter Buffer (MRPB) at the test bus. A publication describes via which driver, channel and data block a message is transmitted from the external device to the system running the ARTIS kernel. A subscription describes how messages received from the test bus can be encoded in a data block and transferred to the external device. This also includes the specification of a driver, channel and data block header. One hardware component may use as many drivers, channels and data blocks as are required to communicate with the external device.

On initialisation of the test configuration, the ARTIS kernel will dynamically link and initialise all desired drivers. MODEST (fig. 8) contains the following hardware components:

£   HW_Control_Panel, to provide the 4096 Code and Altitude Report Enable to the transponder;

£   HW_Aircraft_Config, to provide the transponder with the airspeed category and the ICAO address;

£   HW_Interrogator, to exchange Mode S data between the transponder and the simulated ground station;

£   HW_Aircraft_Systems, to provide altitude information and the air/ground discrete to the transponder;

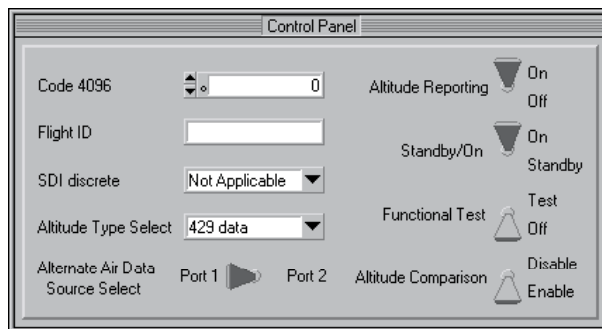£   HW_Arinc718, to exchange messages between the simulated ADLP and the transponder.

Figure 7 Control Panel Stub Component

Stub components describe the graphical user interface interaction with the test bus. Its subscriptions determine where and how which messages in the MRPB are displayed to the operator. Its publications describe how the operator can inject messages into the MRPB. MODEST contains the following stub components:

£   Stb_Control_Panel (fig. 7);

£   Stb_Aircraft_Config;

£   Stb_Interrogator_uplink;

£   Stb_Interrogator_downlink;

£   Stb _Aircraft_Systems

£   Stb_Arinc718_uplink

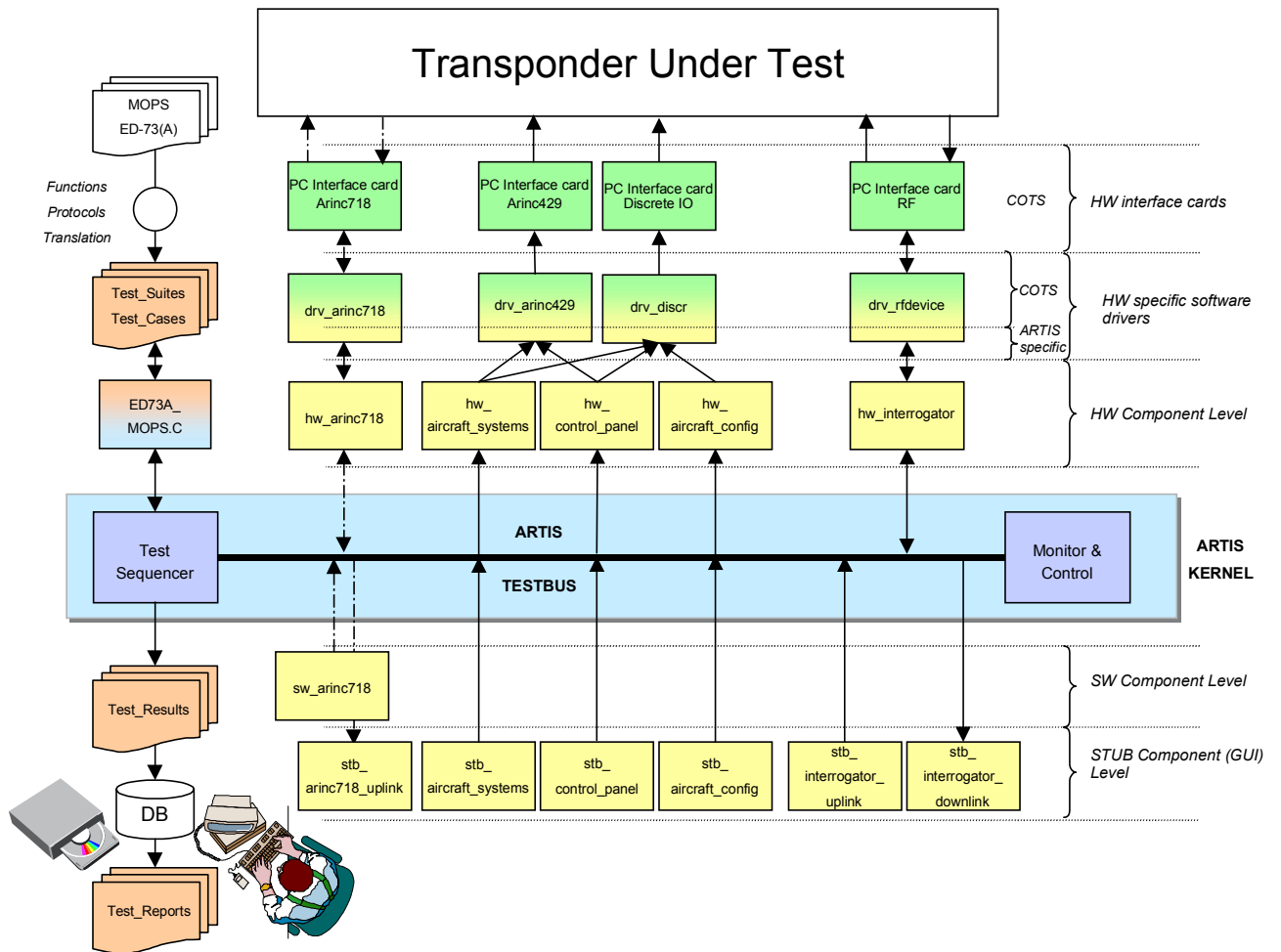Figure 8 MODEST Software architecture

excerpt from ED-73A section 5.4.12.2.2

c.   STEP 3 - Pressure Altitude (Mode S)
Connect the transponder RF port to the transponder test set.

Interrogate the transponder with a standard surveillance-altitude interrogation (UF=4) with the PC, RR, DI, and SD fields set to ZERO and the address the same as that provided to the transponder.

With the ALT switch ON, provide altitude code inputs to the transponder which should result in setting each of the AC field bits (including the M bit or the Q bit) of the reply, one at the time.

Record that the correct bits are transmitted in the AC field of the reply.
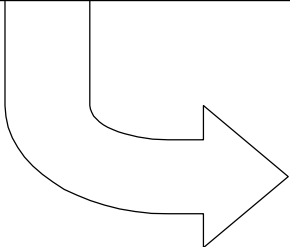
```
/* excerpt from 5_4_12_2_2.c */

int AC[10]={-1000, -800, -500, 500, 2500, 6500, 14500, 30500, 62500, 126500};
Progress("STEP 3 - Pressure Altitude (Mode S)");
  /* ------------------ stimulus ------------------------------------------------- */
memcpy(&stimulus, &ALTREPRT, sizeof(ALTREPRT)); stimulus.AltReprt = 1; //
enable altitude Processing send (P_ALTREPRT, &stimulus, NOW);

for (i = 0; i < 10; i++)  // FZ: only 10 patterns are required
  {
  // enter altitude code into transponder
  /* ------------------ stimulus ------------------------------------------------- */
  memcpy(&stimulus, &ALTITUDE, sizeof(ALTITUDE));
  stimulus.Altenabl = 0;          // use normal altitude Processing
  stimulus.Altitude = AC[i];
  send (P_ALTITUDE, &stimulus, NOW);
  Wait (1); /* Wait [s] */

  // extract altitude reply
  /* ------------------ stimulus ------------------------------------------------- */
  memcpy(&stimulus, &SURVALTI, sizeof(SURVALTI));
  stimulus.UF = 4;
  stimulus.PC = 0;
  stimulus.RR = 0;
  stimulus.DI = 0;
  stimulus.IIS = 0;
  stimulus.SDuse = 0;
  stimulus.SD = 0;
  stimulus.AD = acaddr;
  send (P_GRSTXPDR, &stimulus, NOW);

  /* ------------------ observation --------------------------------------------- */
  memcpy(&observation, &SURVALTR, sizeof(SURVALTR));
  timeout = Timeout;
  res = receive (P_XPDRGRST, &observation, timeout, &rectime);
  if (res == TIMEOUT)
    fail("Transponder did not reply to UF 4");
  else if (observation.AC != AC[i]) {
    sprintf (reportstring, "Transponder did not deliver correct altitude code
(observed: %d, expected: %d)", observation.AC, AC[i]);
```

Figure 9 From Test Procedure to Test Case

```
0000052754; H0; 5_4_12_2_2_20020109_114309;
0000052754; H2; 20020109;
0000052802; TC; performTest 5_4_12_2_2;
0000052802; CS;   MODEST Test Script - National Aerospace Laboratory NLR
  generated by Ms-Access on 19/9/01 at 17:09:50
  file: H:\modest\II - Projekt Fase\II.C Technical\II.C.4 SE Interfaces\tmp\scripts\5_4_12_2_2.tst
  WARNING: Do not modify this file! It is computer generated, modification will cause inconsistencies!
  Variable Direct Data  This test verifies that the transponder replies the correct data as applied to
  the variable direct data interfaces (Pressure Altitude, 4096 Identification Code, FS and VS for Mode-S).;
0000053798; H4; Honeywell-xs950;
0000053888; CC; STEP 3 - Pressure Altitude (Mode S);
0000053897; DC; ALTREPRT [AltReprt: 1];
0000053907; DC; ALTITUDE [AltEnabl: 0, Alt: -1000];
0000054918; DC; GRSTXPDR [UF: 4, PC: 0, RR: 0, DI: 0, IIS: 0, MBS: 0, MES: 0, LOS: 0, RSS: 0, TMS: 0, TCS: 0, RCS: 0, SAS: 0, SIS: 0, LSS: 0, RRS: 0, SDuse: 0, SD: 0, AD
0000054960; DO; XPDRGRST [DF: 4, FS: 0, DR: 0, IIS: 0, IDS: 0, AC: -1000, AP: 38019b] at 54921 ms;
0000054972; DC; ALTITUDE [AltEnabl: 0, Alt: -800];
0000055989; DC; GRSTXPDR [UF: 4, PC: 0, RR: 0, DI: 0, IIS: 0, MBS: 0, MES: 0, LOS: 0, RSS: 0, TMS: 0, TCS: 0, RCS: 0, SAS: 0, SIS: 0, LSS: 0, RRS: 0, SDuse: 0, SD: 0, AD
0000056067; DO; XPDRGRST [DF: 4, FS: 0, DR: 0, IIS: 0, IDS: 0, AC: -800, AP: 38019b] at 56000 ms;
-
0000105991; DC; ALTITUDE [AltEnabl: 0, Alt: 126500];
0000107006; DC; GRSTXPDR [UF: 4, PC: 0, RR: 0, DI: 0, IIS: 0, MBS: 0, MES: 0, LOS: 0, RSS: 0, TMS: 0, TCS: 0, RCS: 0, SAS: 0, SIS: 0, LSS: 0, RRS: 0, SDuse: 0, SD: 0, AD
0000107035; DO; XPDRGRST [DF: 4, FS: 0, DR: 0, IIS: 0, IDS: 0, AC: 126500, AP: 38019b] at 107008 ms;

Test-result: PASS, (Finished: 11:44:44)
REPORT of line 10, time 147593
Test-report finished.
```
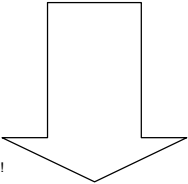
Figure 10 Example Test Result File before Test Report Production

Tool Verification

The components that MODEST is composed of were specified in detail, implemented and tested by a team of software engineers. Individual component testing was conducted in a laboratory environment. Especially the low-level drivers that interface to the physical world were tested outside the ARTIS kernel. During early integration testing these drivers are connected to the ARTIS hardware components and to the test bus. ARTIS also provides means to monitor the parameters on the test bus allowing early verification of the stub components before using them in conjunction with the hardware components. The most complex components are necessary for the ARINC-718 data link interface and the RF interrogator. A dedicated ARINC 718 protocol analyzer was used to monitor the protocol exchange between MODEST and the transponder. In other circumstances dedicated test cases were written to exercise the test system, and to check whether the required ranges can be handled.

After successful completion and validation of the MODEST test engine itself, the compiled test scripts were integrated with the MODEST test engine. Their execution was verified using a set of reference transponders. These transponders have either a track record of good performance or known deficiencies, and should result in a reliable interface with MODEST.

After all test cases were verified, the Formal Tool Verification was performed. Each requirement from the TOR was verified. The report containing the achieved test results, documented remaining deficiencies, together with the Product Baseline configuration management statement, formed the basis for MODEST tool qualification.

First usage test results

At Eurocontrol Experimental Center (EEC) (Bretigny sur Orge, France) the test bench has been set up with the objective to prepare the deployment of Mode S Elementary Surveillance in the core area of Europe. Among other tasks, EEC plays a key role in investigating any potential deficiencies that might prevent the correct operational use of a Mode S system (both transponders and ground stations). Since the installation of MODEST at EEC, a number of reported anomalies have been further investigated and independently confirmed in the laboratory, using the MODEST bench. Transponder anomaly reports were recorded and maintained by EEC. Details can be found at the EEC's website [11].

Future work

After a first delivery of the basic tool set early 2002, an upgrade of the tool with test suites to verify the ACAS-transponder interface in accordance with ED-73A and ARINC-735 is under development. Extensions for testing the transponders' extended squitter functionality for ADS-B and the verification of General Aviation transponders are being considered.

Conclusions

MODEST was developed and delivered for independent verification of multi-vendor Mode S transponders. It has been qualified to meet its operational requirements.

An initial selection of test procedures from the ED-73A MOPS can be automatically executed, taking approx. anywhere between 15 seconds up to 14 hours (depending on the test procedure) to complete. Some improvements of the formal test procedures were identified and proposed to an EUROCAE working group involved in the definition of the requirements and associated verification.

Within the first months after installation of MODEST, the tool has been successfully used to confirm a number of reported Mode S transponder installation anomalies.

Transponder manufacturers have been invited to submit their equipment for independent verification to EEC. All Provisional Fault Reports will be treated confidentially.

References

[1] Aeronautical Telecommunications, Annex 10 to the convention on international civil aviation, volume III Part I - Digital Data Communications Systems, ICAO, 5 November 1998

[2] Aeronautical Telecommunications, Annex 10 to the convention on international civil aviation Annex 10 2nd edition of volume IV Surveillance Radar and Collision Avoidance System, ICAO, 5 November 1998

[3] Mode S Programme, Air Traffic Surveillance, Eurocontrol, http://www.eurocontrol.int/mode_s/

[4] ARINC, Mark 3 Air Traffic Control Transponder (ATCRBS/Mode S), ARINC Characteristic 718-4, December 15, 1989

[5] Manual on Mode S Specific Services, ICAO, Doc 9688

[6] RTCA, Minimum Operational Performance Standards for Air Traffic Control Radar Beacon System/Mode Select (ATCRBS/Mode S) Airborne Equipment, DO-181C

[7] EUROCAE, February 1999, Minimum Operational Performance Specification for Secondary Surveillance Radar Mode S Transponders, ED-73A

[8] RTCA, 1992, Software Considerations in Airborne Systems and Equipment Certification, RTCA Document DO-178B

[9] Aartman, L.J., L.C. Eveleens, S. Wellink, "Avionics Development in a concurrent engineering environment, from virtual prototyping to testing", NLR-TP-98135, July 1998, http://www.nlr.nl/public/library/1998/98135-dcs.html

[10] Information technology – Open Systems Interconnection – Conformance testing methodology and framework, ISO/IEC 9646-1, second edition, 1994-12-15.

[11] Transponder Anomalies, Mode S Programme, Eurocontrol, http://www.eurocontrol.int/mode_s/Monitoring/airborne_monitoring.htm