# INTRODUCTION OF SATELLITE COMMUNICATION SYSTEM TEST LANGUAGE

**Dong Hyun Ahn and Won Chan Jung**

**Satellite Communication Network Control Section**
**Satellite Communications Division, ETRI**
**Yusong P.O.Box 106, Taejon 305-600, Korea**
**Tel.(042)-860-6479, Fax. (042)-860-6430**

*ABSTRACT*

For the effective use of satellite communication transponder, tests for the payload system such as IOT (In-Orbit Test), RPM (Routine Payload Monitoring), CSM (Communications System Monitoring), and REV (Remote Earth-Station Verification) have to be conducted. Those tests are used in order to verify the condition and generic design of the satellite, to provide a database for operational calculations, and to maintain the quality of communication services. As the satellite communication system gets the wider expansion with higher complexity of operation, tests for the communication system also need more complex operation that use sophisticated computer-controlled measuring system. For this purpose, there is a set of SICL library which is a BASIC and C language based measurement functions, which uses GP-IB protocol and SCPI commands. But SICL requires knowledge of BASIC and C language as well as GPIB and SCPL system[1],[4].

This paper introduces a new language called CALSTEP - Control and Access Language for the Systems of Test Equipment and Payload. This language is designed for the operator to perform the tests for the satellite communication system without any special knowledge that is mentioned above. This language has very limited number of commands which are to be used to control the payload system and test equipments to perform IOT and CSM, and those commands are very readable and easy to understand, so an operator without any knowledge of BASIC and C programming language, or SICL and SCPI command can use it.

## I. Introduction

The role of satellite communications becomes more and more important in the field of communications as the satellite and communication technology is developed rapidly. To open and maintain the communication path via satellite, it is necessary to test the payload system before and during the communication service is provided. For this purpose, several types of tests are performed, and those are IOT (In-Orbit Test), RPM (Routine Payload Monitoring), CSM (Communication Systems Monitoring), and REV (Remote Earth-Station Verification) [3].

Even though the payload system has been tested for many times on the ground, it is always possible that the communication system in the satellite may be damaged during the launch time. Therefore, after the satellite is located at the proper position, it is desirable to test the payload system over again to check if the system is ready to be used. IOT is the test for this purpose. After the IOT is completed, and the service is provided, the payload system has to be monitored regularly for any possible deficiency, and that regular monitoring is called RPM. Along with RPM, the communication service quality via satellite is also tested to see if the communication between earth stations via satellite is operating correctly. This monitoring is called CSM. Finally, it is desirable to check the earth stations which are communicating via satellite to prevent illegal and harmful signal transmission to the satellite. That is why REV is performed.

Those tests for the payload system in space are performed from the ground using several RF test equipments to test several parameter measurements. For those tests, telecommand (signal to the satellite) is sent, and telemetry (signal from the satellite) is received. Since it is almost impossible to perform these tests by controlling the equipments manually, they are usually tested using computer system with pre-defined test procedures and test equipments control commands. One of those pre-defined test command is SICL. A set of SICL library provides those commands that are necessary to build procedures for the tests. However, using SICL to build test procedures is very complicated, and if one wants to perform the test that is not pre-defined using SICL library or perform the part of the pre-defined test, he needs to define the test parameter that he wants to operate. Obviously, it is not only very inconvenient and time consuming to design the test parameter every time the operator wants to perform the test not included in the SICL library, but also is quite hard for the operators without any knowledge of BASIC and C programming language since

SICL library is designed using those languages [4].

CALSTEP is the language born to solve the problem mentioned above. It is a simple language just to be used to perform IOT and CSM with very limited number of statements. So, an operator without any knowledge of BASIC or C language can learn CALSTEP and perform the tests without any problem.

Furthermore, when any set of commands is sent to the transponder, there is always a possibility that a wrong set of telecommands may be sent and damage the transponder by unexpected change of network configuration. CALSTEP provides a function of formal verification of the set of telecommands before it is actually sent to the transponder in order to prevent those undesirable situation.

## II. Language Description

### 1. Design Issue

The main issue in the design of CALSTEP is simplicity, flexibility, and single purpose. Since CALSTEP is made for the operators who do not have any knowledge of BASIC or C programming languages, it is designed to have a limited number of commands just enough to perform IOT, RPM, CSM, and REV tests. Also the language is very flexible, with the help of window environment, because any part of a program can be executed without making another program file or updating the existing program for the execution. This issue of flexibility is very useful when the test is performed because step by step test can be performed for the detail test. Finally, CALSTEP is designed for the communication system tests and only for those tests. Of course it is possible to expand the number of commands in this language and use it for some other tests in the future, but it will remain as a language for the communication system tests.

### 2. Syntax

The program is divided into two parts : setting block and test procedure block. In the setting block, the operator sets the proper parameters for the test. Those parameters are satellite and transponder to be tested, channel number, test name, earth station, carrier, output type, weather condition, etc. This setting block has to be included in every test program in CALSTEP.

In test procedure block which has free formatted structure, any type of test procedure can be designed. For the simplicity, there are only four different statements in CALSTEP : assignment statement, command statement, label statement, and control statement. Assignment statement is used when certain value is calculated and saved in a variable. Command statement generates command to the test equipments or to the payload system. Label statement simply provides the label to be connected from any part of the program. And finally, control statement controls the flow of the program. The example of the program and the result of the program are introduced in section III.

### 3. Running Environment

CALSTEP itself doesn't require any environment except text editor to design the test program. However, since the test program written in CALSTEP is designed to perform IOT/CSM tests, those test equipments and payload system, or at least simulators, are necessary. Also needed is the configuration file for the payload system. The figure 1 shows

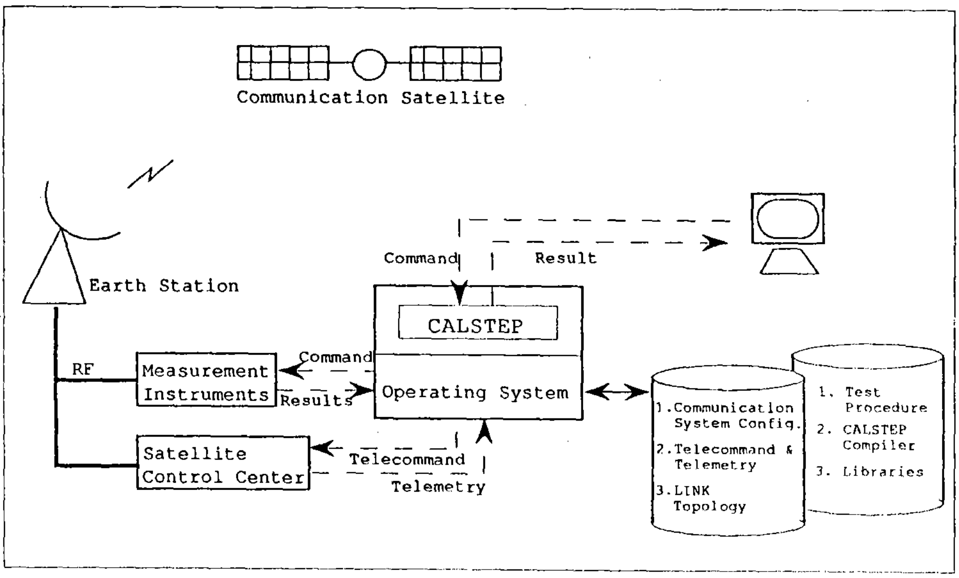


**Figure 1. Running Environment of CALSTEP**

the running environment for CALSTEP.

## III. Examples

### 1. A Sample CALSTEP Program

```
;             <SAMPLE TEST PROCEDURE >
;
;Test Procedure for testing the Input-to-Output power
;Characteristic of ETRISAT Transponder FSS channel 6.
;

#SETTING;                          Begin setting
   satellite = ETRISAT;            Lab. model transponder
   transponder = 6;                FSS
   channel = 6;
   variable   Antenna_Size, Tx_WG_Loss, ...;
            .
            .
            .

#PROCEDURE;                        Begin test

step1;   Import configuration data
   LINK_FREQUENCY(FU,FD);      Read Up/Down link Freq.
   EARTH_STATION_CONFIG(Antenna_Size,Tx_WG_Loss,...);
   LINK_TOPOLOGY(topology);
   TC_TM_CODE(code);

step2;   Declaration of test results
   RESULT(Flux_Density, EIRP);

step3;  Initialization
   scc(LNA_RECV,ON);          Channel 6, LNA RECEIVER ON
   scc(TWTA,ON);              Channel 6, TWTA POWER ON
   scc(LIN,ON);          Channel 6, LINEARIZER POWER ON
   set_sw_position(0,C147,B);    Input_switch position
            .
            .
            .

step4;   Initialization of Instruments
   gpib(path,C,C,A,C,D);          Set measurement path
   gpib(sg1,0,0,0,1);           Synthesizer 1 POWER OFF
   power_level = -20.0;   Synth. Initial Output Power
   gpib(sa1,SPAN,FD,FU,13, power_received, ...);
   gpib(pm1,FU,5,power_transmitted,error);
            .
            .
            .

step5;   Main Routine
   gpib(sg1,FU,output_power,1,1); Synth. 1 Power Gen.
   gpib(sa1,SPAN,FD,FU,2,power_received, ...);
   gpib(mm1,on,voltage,error);
   gpib(pm1,FU,1,power_transmitted,error);
   gpib(sa1,SPAN,FD,FU,3,power_received, ...);
   ATTul = 32;                     Up-Link Attenuation
   Cul = Tx_Coup_Loss + Tx_Cab_Loss + Tx_WG_Loss
       - ATTul;                    Calibration Up-Link
   Gt = 20*log((1/0.2998)*PI*Antenna_Size*fu*0.7416);
       ; Tx Antenna Gain
   SF = 10 * log(4*PI*Range^2); Spreading Factor
   Flux_Density  = power_transmitted - Cul + Gt - SF;
            .
            .
            .

  Volt_diff = Vt_new - Vt_old;

   if (Volt_diff < -0.0005)    jump step6;
   output_power = output_power + power_augment;
   jump step5;

step6;  Test termination & output power OFF
   gpib(sg1,0,0,0,1);           Synthesizer 1 POWER OFF
   gpib(sg2,0,0,0,1);           Synthesizer 2 POWER OFF
```

### 2. Explanation

The CALSTEP program to test ETRISAT Transponder FSS Channel 6 is written and presented in section 3.1. It has a free format except for the documents after semicolon which are treated as comments in CALSTEP. Setting block begins after the keyword **#SETTING**, and the test procedure block begins after the keyword **#PROCEDURE**. Command statements **scc** and **gpib** send the command to the payload system and test equipments respectively. This example is the pre-defined test procedure named gain transfer characteristic [2],[3]. This procedure written in CALSTEP can be executed in several parts which enables more detailed test.

## IV. Conclusions

To provide the qualified satellite communication, more accurate test for the communication system is necessary with more various test methods and more complicated control commands. CALSTEP is a simple, flexible, and specific language which provides the methods of test and monitoring the communication transponder to the operator without any knowledge of programming language or automatic control. Furthermore, it provides the fault-tolerant test environment by verification of the test procedure before it is executed which might be harmful to the communication system.

Future research includes the adaptation of the expert system to the CALSTEP. It requires more knowledge of transponder to build the expert system and various types of test procedure library. Also CALSTEP may be expanded to be used in automatic control system other than satellite transponder.

## Reference


[1] Sara M. Mussmann, "GPIB and the Battle of Incompatible Languages," *IEEE Trans. Instrumentation and Measurement*, vol.37, no.4, pp. 488-492, Dec. 1988.

[2] K.D.Fullett *et al.*, "Microwave measurement system software," *COMSAT Technical Review*, vol.23, no.1, pp. 101-134, Spring 1993.

[3] Arther F. Standing, "Measurement techniques for in-orbit testing of satellites," *Computer science press*, 1990.

[4] HP SICL Reference Manual, *Hewlett Packard Company*, 2nd ed., Dec. 1992.

[5] I. Dostis *et al.*, "In-orbit testing of communications satellites," *COMSAT Technical Review*, vol.7, no.1, pp. 197-225, Spring 1977.