

다목적실용위성 2호의 데이터 획득을 위한 자동적인 포맷 테이블 생성

이재승⁰*, 최종욱*, 천이진*
*한국항공우주연구원

Automatic Generation of Format Table for Data Acquisition for KOMPSAT-2

Jae-Seung Lee⁰*, Jong-Wook Choi*, Yee-Jin Cheon*
*Korea Aerospace Research Institute

요약

최근 우주산업의 급속한 발달로 다양한 목적의 인공위성들이 개발되고 있다. 이러한 인공위성들은 그 사용 목적에 따라 다르지만 매우 많은 양의 데이터들을 다루게 된다. 이러한 데이터에는 각각의 주어진 임무에 사용되는 정보들이 대부분을 차지하지만 위성자체의 상태 데이터도 일정시간마다 확인을 해주어야 한다. 위성의 상태 데이터는 위성 각 부분의 이상유무나 위치, 시간 등 위성이 정상적으로 그 역할을 수행하고 올바른 궤도를 유지하는데 필요한 정보들이다. 위성탑재 소프트웨어는 하드웨어 데이터나 위성의 상태 데이터를 정해진 시간간격마다 읽어 메모리에 저장해 두는 일을 한다. 다목적실용위성 2호에서 이러한 데이터 획득을 자동적으로 수행하기 위해 매 시간간격마다 요구되는 데이터를 테이블 형식으로 헤더파일에 생성하는 프로그램을 구현하였다. 이 구현된 프로그램을 이용하여 다목적실용위성 2호의 탑재소프트웨어가 개발되고 있다.

1. 서론

인공위성을 제어하고 임무를 수행하는 데에는 많은 데이터들이 사용되어진다. 이러한 데이터를 획득하고 저장하는 일련의 과정을 telemetry processing[1]이라고 한다. 본 논문에서는 다목적실용위성 2호의 탑재소프트웨어[2]의 설계과정에서 telemetry processing을 수행하기 위해 사용되어지는 하드웨어 테이블과 포트 테이블, 프레임들을 자동적으로 생성하는 프로그램 구현에 대해 설명한다. 데이터 획득 및 저장시 참조하게 되는 이러한 테이블들은 하나의 헤더파일로 만들어지게 되는데 필요한 데이터들이 정의된 DB 파일로부터 이 헤더파일을 자동적으로 생성하는 프로그램도 위성탑재소프트웨어의 한 부분으로 포함된다고 볼 수 있다. 본 설명에 들어가기 전에 다목적실용위성 2호에서 telemetry processing이 수행되는 전체적인 흐름에 관하여 설명하고 각각의 테이블들이 수행과정에서 어떻게 사용되며, 이러한 테이블들을 생성하는 알고리즘에 대해 알아본다.

2. 위성탑재소프트웨어의 Telemetry Processing

다목적실용위성 2호의 telemetry processing은 탑재소프트웨어의 CSCs(Computer Software Components) 중 DAQ(Data Acquisition)에 의해서 수행된다. 표 1은 DAQ의 데이터 구조를 나타낸다.

Structure	Description	OBC	RDU	ECU
Master Hardware Table	Contains hardware access information for all HW devices	1	1	1
Master Telemetry Table	Contains GDA access information for all telemetry variables	1	1	1
Application Port Table	Contains master hardware table indices for acquisition of hardware items needed by non-daq software	4	4	4
Telemetry Port Table	Contains master hardware table indices for acquisition of hardware items needed for telemetry that were not acquired by the application port table(s)	3	3	3
Mejor Frame Table	Contains master telemetry table indices for acquisition of all telemetry items	3	3	3
Telemetry Buffer	Contains telemetry data for current minor frame	1	1	1

표 1. DAQ Data Structure

DAQ는 매초마다 256 바이트의 telemetry를 VCDU(Virtual Channel Data Unit) 포맷의 minor 프레임으로 저장하며 총 32개의 minor 프레임이 존재하므로 32초마다 같은 minor 프레임이 반복되며 이를 major 프

레이미라 한다. DAQ는 이러한 minor 프레임에 저장될 데이터뿐만 아니라 다른 CSC에서 요구하는 데이터의 획득도 담당한다.

2.1 다른 application에 대한 하드웨어 데이터 획득

탐재소프트웨어에서 DAQ를 제외한 다른 CSCs에의 해 요구되는 하드웨어 데이터는 MHT(Master Hardware Table)와 APT(Application Port Table)을 이용하여 얻어진다. APT는 각 모듈의 수행시간에 따라 표 2에 제시된 4가지가 존재하며 그림 1은 APT를 이용한 하드웨어 데이터 획득과정을 나타낸다.

APT	Description	Applications
QTR	run every 1/4 second	ACS, MMD
ONE	run every second	ACS, EPS
SXT	run every 16 seconds	ACS, EPS
INIT	during initialization	EXC

표 2. Application Port Tables

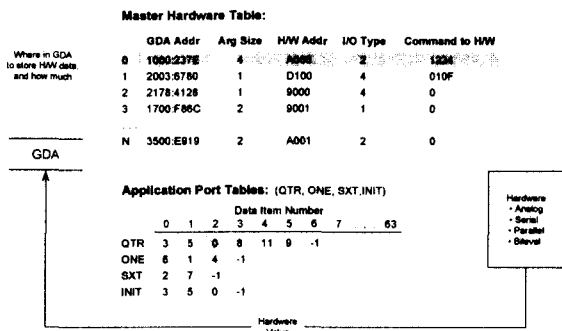


그림 1. Hardware Data Acquisition

2.2 DAQ에 대한 하드웨어 데이터 획득

DAQ는 2.1절에서 획득되지 않은 데이터에 대해서만 현재의 minor 프레임에 대한 하드웨어 데이터를 얻을 필요가 있다. 획득방식은 2.1절과 동일하게 look-up 테이블을 이용하는데 APT대신에 TPT(Telemetry Port Table)이 사용된다는 점이 다르다. 현재의 minor 프레임번호가 TPT의 열을 선택하게 되고 TPT의 각 열에 있는 값은 데이터의 주소정보를 가진 MHT(Master Hardware Table)의 인덱스를 가르킨다. DAQ는 TPT를 참조하여 MHT가 나타내는 주소의 데이터를 가져와 버퍼에 저장하기만 하면 된다.

2.3 VCDU 및 telemetry segment의 생성

VCDU를 생성하기위한 하드웨어 데이터는 2.2절의 방법에 의해 구해진다. 그리고, VCDU 헤더와 프레임 헤더를 생성하고 그림 2에 나타난 것과 같이 MFT(Major Frame Table)을 상요하여 telemetry 데이터를 버퍼에 저장하게 된다. 만약 버퍼에 남은 영역이 존재하면 지상에서 보낸 명령에 의해 덤프 데이터로 채워지게 된다.

SOH를 포함한 telemetry 데이터의 획득도 2.2절에서 설명한 TPT를 이용한 하드웨어 데이터 획득과 유사한 방식으로 이루어진다. 먼저 현재의 minor 프레임 번호가 MFT의 열을 선택하고 MFT의 각 열은 메모리의 telemetry 값을 읽어서 버퍼에 저장하기위한 MTT(Master Telemetry Table)의 인덱스 값을 가지고 있다. VCDU telemetry 버퍼의 크기는 220 바이트로 고정되어 있으며 sync-pattern과 RS-code를 합치면 전체 VCDU의 크기는 256 바이트가 된다. 그림 3은 telemetry의 획득과 버퍼에 저장하는 과정을 보여준다.

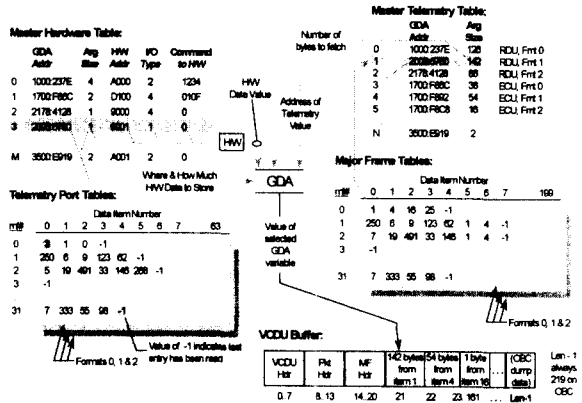


그림 2. Telemetry Acquisition

3. Telemetry 포맷 테이블 생성 프로그램

생성해야할 테이블들은 MHT, APT, MTT, MFT, TPT의 순서로 만들어진다. 이 테이블들은 다른 모듈들에서 참조할 수 있어야 하므로 헤더파일로 생성되어야 한다.

입력으로 사용되는 데이터베이스 파일들은 지상에서 수신해서 확인할 필요가 있는 위성의 상태 데이터들을 미리 정의해 놓은 것으로 MHT와 APT의 생성을 위한 파일, MTT의 생성을 위한 파일, 그리고 MFT와 TPT의 생성을 위해 각 minor 프레임의 정보를 가지고 있는 파일들이 필요하다.

그림 3과 4는 각각 하드웨어 데이터베이스 파일로부터 MHT 및 APT를 생성하는 알고리즘과 이때 사용되는 DB 파일과 생성되는 테이블들의 일부분으로 각 테이블들의 형식을 나타낸다.

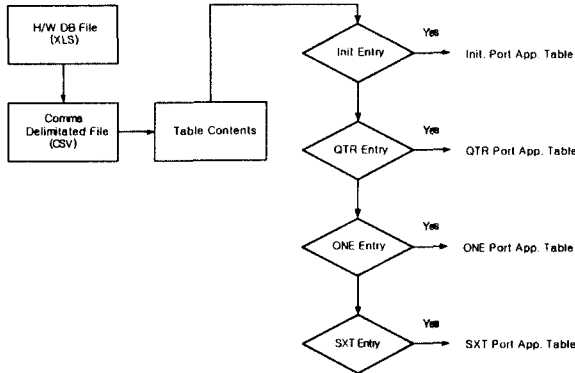


그림 3. Flow Diagram of Port Table Generation

RDU Hardware Data Base

	C Variable	Size	I/W Addr.	cmd to hw	INIT	QTI	ONF	SX	TPT ref
10	SP_rdu_parallel_tlm[10]	2	0x7014	0x0000	na	na	na	na	yes
11	ACS_gyro_counts[0]	2	0x9000	0x0000	na	QTI	na	na	no
29	SP_rdu_parallel_tlm[13]	2	0xF200	0x0000	na	na	na	na	yes
30	SP_rdu_bilevel_tlm2	2	0xB100	0x0000	INIT	na	ONF	na	no
43	SP_rdu_analog_2b_tlm[0]	2	0xA200	0x1A0A	na	na	na	SX	no

Master H/W Table (MHT)

```
struct KPD_master_hw_tbl_struct KPD_master_hw[SP_MASTER_HIW_SIZE] =
{
/* 10 *//(unsigned char *)&SP_rdu_parallel_tlm[10], 2, 0x7014, 0x0000),
/* 11 *//(unsigned char *)&ACS_gyro_counts[0], 2, 0x9000, 0x0000),
/* 29 *//(unsigned char *)&SP_rdu_parallel_tlm[13], 2, 0xF200, 0x0000),
/* 30 *//(unsigned char *)&SP_rdu_bilevel_tlm2, 2, 0xB100, 0x0000),
/* 43 *//(unsigned char *)&SP_rdu_analog_2b_tlm[0], 2, 0xA200, 0x1A0A),
};
```

Init. Application Port Table (APT)

```
struct KPD_app_port_table_struct KPD_init_port_tbl =
{
/* 30 *//(unsigned char *)&SP_rdu_bilevel_tlm2 RDU/OBC DPLL Status */
-1
};
```

그림 4. H/W Database & MHT & APT

MFT와 MTT를 생성하기위해 먼저 Telemetry 데이터베이스로부터 각 telemetry를 얻을 수 있는 주소와 이름을 이용하여 MTT를 생성하고 매 초마다 minor 프레임에 필요한 telemetry를 Grid 파일로부터 읽어서 이에

해당하는 MTT의 인덱스를 이용하여 MFT를 생성한다. 그림 5는 생성된 MTT와 MFT의 형태를 나타낸다.

Master Telemetry Table (MTT)

```
struct KPD_master_tlm_tbl_struct
KPD_master_tlm[SP_MASTER_TLM_SIZE] =
{
/* Row C Variable Bytes
/* 50 *//(unsigned char *)&SP_obc_bilevel_tlm[0], 2),
/* 51 *//(unsigned char *)&SP_obc_bilevel_tlm[1], 2),
```

OBC Major Frame Table (MFT)

```
{
/****** MINOR FRAME # 3 *****/
/* MTT Byte Grid Mom */
/* ROW Size Index C Variable Mnemonic */
/*-----*/
50, /* 2 213 SP_obc_bilevel_tlm[0] CTCRSBL1*/
-1, /* 209 FIRSTMINUSONE */
-1, /* 209 LASTMINUSONE */
/*******/
```

그림 5. Structures of MTT & MFT

4. 결론

본 논문에서는 telemetry 데이터를 획득하기 위한 테이블이 정의되는 헤더파일을 자동적으로 생성되는 프로그램을 설명하였다. 하나의 위성에는 여러 개의 프로세서가 장착되며 각각의 프로세서에서 필요한 정보에 대한 DB 파일만 제공하면 각각의 테이블들은 자동적으로 생성된다. 생성된 헤더파일들은 다목적실용위성 2호의 탑재 소프트웨어에서 사용되기 위해 검증 시험을 하고 있다.

참고문헌

- [1] 강수연 외, "아리랑 위성의 Command/Telemetry 시스템", 한국정보과학회 가을 학술발표논문집(III), 제 25권 2호, pp.662-664, 1998.
- [2] 이종인 외, "아리랑 위성 탑재 소프트웨어 소개", 한국정보과학회 봄 학술발표논문집(A), 제 25권 1호, pp.741-743, 1998.