위성 운용을 위한 조건형 순차 명령 집합 설계

신현규*
*한국항공우주연구원
e-mail : hkshin@kari.re.kr

A Design of Sequence Command Set with Condition for Satellite Operation

Hyun-Kyu Shin*
*Korea Aerospace Research Institute

요 약

기존의 저궤도 위성에서는 RTCS 이용하여 순차적인 명령 집합을 구현해왔다. 장치의 초기화, 관련 모드의 변경, 임무 수행 등 많은 부분에서 연관된 순차 명령이 실행되도록 설계되어 적용되었다. RTCS 는 연관된 순차 명령을 정해진 시간 간격을 두고 실행 할 수 있도록 해주었다. 이러한 RTCS 의 수행 중, 위성의 상태에 따른 조건 판단이나 분기가 필요한 경우, 위성비행소프트웨어에 별도의 Logic 을 구현하였다. 이 방법은 RTCS 자체의 구조가 단순하고, 그 수행이 직관적이라는 장점이 있으나, RTCS 와 관련된 Logic 이 위성비행소프트웨어에 고정적으로 구현되고, 조건 판단 및 분기가 여러 부분에서 수행될 경우 RTCS 가 단편화되며, 위성비행소프트웨어에서 구현해야 하는 Logic 의 관계가 복잡해지는 단점이 있다. 본 논문에서는 기존 RTCS 가 가지는 단순한 구조를 유지하면서, 조건 판단 및 분기의 처리가 가능한 새로운 순차 명령 집합의 설계에 대해 소개한다.

1. 서론

위성을 운영하기 위해서는 위성의 행위를 유발하는 다양한 명령이 전달되어야 한다. 위성에 탑재되는 위성비행소프트웨어는 이 명령을 받아 사전에 정의된 동작을 수행함으로써 위성을 제어, 임무를 수행한다. 이러한 위성의 운용에는 순차적으로 전달되어야 하는 명령들이 존재한다. 예를 들어 특정 장치를 초기화하기 위해서는 장치가 작동할 수 있는 온도 조건을 맞추고, 전원을 켜고, 초기값들을 설정한 후, 운용 모드로 전환하는 등의 절차가 필요하다. 이러한 명령들은 지상에 의해서 순차적으로 전송될 수 도 있지만, 미리 정의된 순차 명령 집합에 의해 처리될 수 도 있다. 기존의 저궤도 위성에서는 RTCS(Relative Time Command Sequence)를 이용하여 이러한 순차 명령 집합을 구현해 왔다. RTCS 는 하나의 명령과 다음 명령의 실행까지 특정 시간 간격(Delta-Time)을 갖는 명령의 집합으로, 위성비행소프트웨어 내에 탑재되거나 또는 지상의 명령에 의해 새롭게 적재될 수 있다. RTCS 의 수행 중, 조건에 대한 판단 및 분기가 필요한 경우에는 위성비행소프트웨어 내에 별도의 Logic 을 구현하고, RTCS 가 해당 Logic 을 Enable 한 후, 관련 Logic 에서 판단을 수행, 필요 시, 조건에 맞는 다른 RTCS 를 Activation 하였다. 이러한 방법은 위성비행소프트웨어 내에 연관 코드를 구현함으로써 RTCS 자체의 구조를 단순화하고 그 수행이 직관적이라는 장점이 있으나, RTCS 의 수행과 관련된 Logic 이 위성비행소프트웨어 내에 고정적으로 구현된다는 점, 조건 및 분기가 여러 부분에서 수행될 경우, RTCS 가 단편화 되며 위성비행소프트웨어에서 구현해야 하는 Logic 의 관계가 복잡해지는 단점을 가지고 있다. 본 논문에서는 기존 RTCS 가 가지는 단순한 구조를 유지하면서, 이러한 단점을 보완하는 새로운 조건형 순차 명령 집합의 설계에 대해 소개한다.

2. 조건형 순차 명령 집합의 요구사항

새롭게 설계하는 조건형 순차 명령 집합에 대한 주요 요구 사항은 다음과 같다.

- Delta-time 으로 연결된 순차 명령의 실행을 보장하여야 한다. (기존 RTCS 의 속성)
- 특정 조건이 만족함을 기다릴 수 있어야 한다.
- 특정 위치로의 분기가 가능해야 한다.
- 특정 저장 공간(변수)를 이용하여 간단한 Logic 의 구현이 가능해야 한다.
- 실행 상태를 변경할 수 있어야 한다.
- 순차 명령의 집합에서 시작 위치를 변경할 수 있어야 한다.
- 다른 순차 명령 집합을 Activation 할 수 있어야 한다.
- On-board 에서의 수정이 용이해야 한다.

3. 접근 방안

조건 판단 및 분기를 위하여 간단한 연산을 포함하

는 Op Code 1 byte 를 추가한다. 기존의 RTCS, 즉 Delta-time 을 포함하는 Command 는 "CMD" Op Code 를 갖는다. 조건 비교 및 간단한 산술 연산, 대입문을 위한 Op Code, 이에 대한 Instruction Field 를 정의한다.

순차 실행을 보장하기 위하여 조건 판단 등, 순차 명령이 아닌 Op Code 의 위치까지는 기존 RTCS 가 ATC 로 변환되는 방식을 그대로 이용하여 각 영역별 순차 실행을 보장한다. 이때, 기본적으로 ATC 로 변환된 마지막 명령의 예상 실행 시간까지 해당 순차명령 집합은 Pending 되도록 한다.

특정 위치로 분기하기 위해서는 순차 명령 집합을 서술하는 단계에서 Label 을 정의하고, 분기하는 경우 Label 을 기술하도록 한다. 이 정보는 Compile 단계에서 실제 Op Code 의 위치로 재해석 된다.

간단한 Logic 구현을 위한 특정 저장 공간은 헤더 부분에 4 개의 저장소를 배치하여, 조건형 순차명령집합에서 내부 Counter 등으로 사용될 수 있도록 한다.

조건 판단의 경우, 해당 조건은 대부분 위성비행소프트웨어 내의 변수 또는 특정 Telemetry 의 값을 이용하므로, 이를 참조하는 형식으로 하고, 순차 명령 집합을 작성하는 단계에서는 변수 이름 또는 Mnemonic 을 기술하고, Compile 단계에서 실제 주소로 변환된다.

다른 순차 명령의 Activation 은 기존의 RTCS 가 다른 RTCS 를 Activation 할 수 있었던 것처럼 Activation 명령을 동일하게 사용한다.

4. 설계

조건형 순차명령 집합의 전체적인 흐름은 그림 1과 같다.

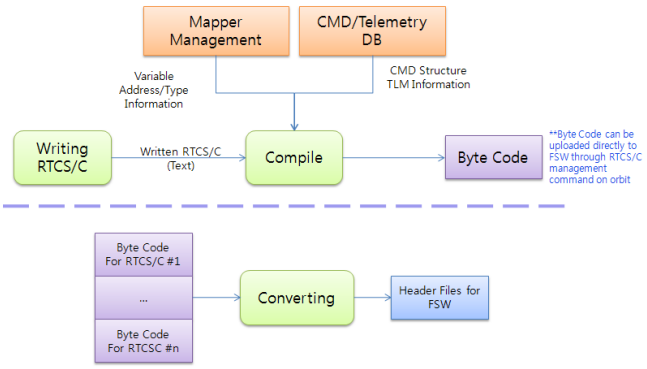

그림 1 Overall Flow

텍스트 형태로 작성된 조건형 순차 명령 집합은 Compile 과정을 거쳐 Byte Code 형태로 만들어진다. 이 때, 위성비행소프트웨어 내의 변수 또는 Telemetry Mnemonic 에 대해서 실제 주소로의 변환이 이루어지며, Label 에 대한 분기도 실제 OpCode 의 위치로 변경된다. 개별적으로 변환된 Byte Code 는 지상 명령에 의해 위성으로 전송될 수 있으며, 지상에서는 별도의 과정을 거쳐 위성비행소프트웨어에 포함될 수 있다.

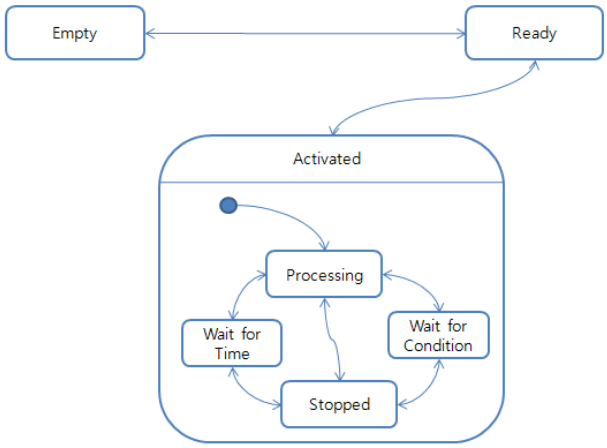

그림 2 State Diagram

그림 2 는 조건형 순차명령 집합의 State Diagram 을 나타내는 것으로 Empty, Ready 및 Activated 의 상태를 갖는다.

- Empty: 아직 조건형 순차 명령 집합이 로드되지 않은 상태
- Ready: 지상 명령 또는 소프트웨어 빌드를 통해 조건형 순차 명령 집합이 적재된 상태
- Activated: Activation 명령에 의해 수행중인 상태

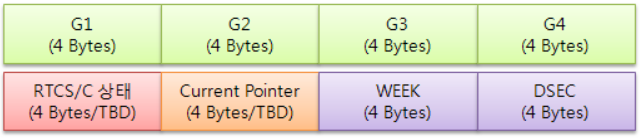

그림 3 Header

그림 3은 조건형 순차 명령 집합의 헤더를 나타낸 것으로 G1~G4 는 Logic 구현을 위한 내부 저장 공간이다. RTCS/C 상태는 현재의 조건형 순차 명령 집합의 상태를 나타내는 것으로 Scheduler 에 의해 참조된다. Current Pointer 는 현재의 수행위치를 나타내며, 이후의 WEEK 및 DSEC 은 수행과 관련된 시간 정보를 저장하게 된다. Command 가 ATC 로 변경되는 경우 마지막 Command 의 실행시간 (Absolute Time)을 기록하게 되며, 조건문 등의 경우에는 해당 처리가 끝난 시간을 기록한다.

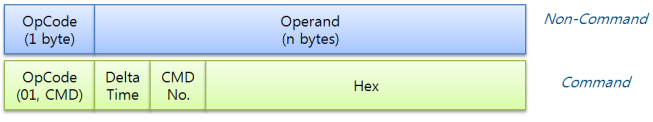

그림 4 Byte Code Format

조건형 순차 명령 집합의 Byte Code 는 그림 4와 같은 형태로 이루어진다. 기존의 RTCS 의 형태를 그대로 유지하면서 그 앞에 Op Code 1 byte 가 추가되었다. Command 의 경우 "01" 의 Op Code 를 가지며, 조건이나 분기, 기타 연산을 위한 Op Code 는 Non-Command

의 형태로 처리된다.

그림 5 에서와 같이 Op Code 는 매우 간단한 연산 및 조건 판단이 가능한 형태로 설계, 확정 예정이며 필요한 경우 Built-in Function 을 제공하도록 Op Code 를 확장할 수 있다.

- **SET** : 저장 공간 (G1~G4)에 특정 값을 설정
- **INC** : 저장 공간 (G1~G4)의 값을 정해진 만큼 증가
- **Condition**
  - CG: X가 Y보다 클때까지 Pending
  - CGE: X가 Y보다 크거나 같을때까지 Pending
  - CE: X가 Y와 같을 때가지 Pending
- **Branch**
  - B: 무조건 분기 (Goto)
  - BG : X가 Y보다 큰 경우 분기
  - BGE : X가 Y보다 같거나 큰 경우 분기
  - BE: X가 Y와 같은 경우 분기
- **Wait** : 주어진 시간 만큼 Pending.

그림 5 Op Code Example

조건형 순차 명령 집합의 Scheduling 방안은 다음과 같다.

- 매 Minor Cycle 마다 각 명령 집합의 상태를 확인
- Wait For Time 의 경우, 해당 시간까지 해석을 수행하지 않음
- Wait For Condition 의 경우, 현재 Current Pointer 의 내용을 수행함.
- 연속된 순차 명령에 대해서는 한 번에 ATC 로 변환. 기존과 동일한 방식으로 연속된 순차명령에 대해서는 각 명령간의 Delta-Time 을 보장함.
- 연속된 순차 명령이 ATC 로 변환되면, 가장 마지막 Command 의 수행시간을 Header 에 기록하고 Wait For Time 으로 상태를 변경함.
- 조건문에 의해 Pending 되는 경우, Wait For Condition 이 되며, 다음 Minor Cycle 에 해당 조건을 다시 판단하게 됨.
- 한번에 실행 가능한 Non-command Op Code 의 수를 제한하여 Starvation 및 Task Overrun 을 고려.

5. 적용 예제

이상에서 설계된 조건형 순차 명령 집합을 이용하여 특정 상황을 가정하여 실제 명령 집합을 구성하였다.

<가정>

이 조건형 순차 명령 집합은 특정 Equipment 를 초기화하고, Normal Mode 상태에서 Equipment 에 포함된 1,2,3 세 개의 장치에 대한 모니터링을 수행. 문제 발생시 #7 순차 명령 집합을 호출하고 종료하도록 설계

<주요 동작>

- Normal Mode 설정을 위해
  - CCMD001 을 주어진 Argument 에 따라 전송
  - CSETANORM 으로 Normal Mode 로 설정
  - Mode 가 Normal 이 될 때까지 기다림
- Monitoring
  - CCAACQDATA 에 해당 장치 번호를 설정하여 명령을 전송
  - Data Acquisition 에는 최대 100 DSEC 이 걸리므로, 다른 조건 판단 없이 100 DSEC Wait.
  - 이후 Acquisition Status 및 Max Value 를 확인하여 문제가 존재할 경우 #7 명령 집합을 호출
  - 정상적일 경우 장치를 바꿔가며 동일 동작을 수행

이를 조건형 순차 명령 집합으로 표현하면 그림 6 과 같다.

```
1  _INIT:
2        SET G1, 100;
3        SET G2, 1;
4        ALIAS G2, DeviceNumber;
5
6  _CHANGE_TO_NORM:
7        CMD 20, CCMD001, A1=0x01, A2=0x02;   // Sending Command 1 with 0102
8        CMD 20, CCMD001, A1=0x02, A2=0x03;   // Sending Command 1 with 0203
9        CMD 25, CCMD001, A1=0x03, A2=0x04;   // Sending Command 1 with 0304
10       CMD 20, CSETANORM;                   // Setting A as Normal Mode
11       CE  TCAMODE, "NORM";                 // Wait until TACMODE is NORM
12
13 _MONITORNG_START:
14       BE DeviceNumber, 2, _DEVICE_2;
15       BE DeviceNumber, 3, _DEVICE_3;
16
17 _DEVICE_1:
18       CMD 10, CCAACQDATA, A1=0x01;         // Start Acquisition #1
19       B   _DATA_CHECK;
20
21 _DEVICE_2:
22       CMD 10, CCAACQDATA, A1=0x02;         // Start Acquisition #2
23       B   _DATA_CHECK;
24
25 _DEVICE_3:
26       CMD 10, CCAACQDATA, A1=0x03;         // Start Acquisition #3
27
28 _DATA_CHECK:
29       WAIT 100;                            // Data Acquisition takes 100 DSEC
30       BNE  TCAACQSTS, "CMPL", _TRIGGER_C;  // If Acquisition Status is not "CMPL", trigger another RTCS/C
31       BG   TCCACQMAX, G1,     _TRIGGER_C;  // If Max Value is greater than G1 (=100), trigger another RTCS/C
32       INC  DeviceNumber, 1;                // DeviceNumber++;
33       BNE  DeviceNumber, 4, _MONITORING_START;
34       SET  DeviceNumber, 1;                // Roll-over Case. This equals with SET G2, 1
35       GOTO _MONITORING_START;              // equals with B _MONITORING_START
36
37 _TRIGGER_C:
38       CMD 10, CCEXECRTCS, A1=0x07;
39
40 _END:
41     TERM;
```

그림 6 작성 예제

6. 결론

기존의 순차 명령 집합인 RTCS 의 단순한 구조는 그대로 유지하고, 1 바이트의 Op Code 를 추가함으로써 조건 판단 및 분기 등의 간단한 Logic 이 가능한 조건형 순차 명령 집합을 설계하고, 이에 대한 적용 예제를 살펴보았다.

현재는 설계 초기 단계로 실제 위성 운용과 관련한 보다 상세한 요구사항에 대한 연구를 진행 중에 있다. 더불어 조건형 순차 명령 집합을 구성하기 위한 전체 시스템의 각 부분에 대한 Proto-type 을 통해 적용 가능성 및 구현에 대한 상세 방안을 연구해 나갈 계획이다.