



## Journal of Knowledge Information Technology and Systems

ISSN 1975-7700 (Print), ISSN 2734-0570 (Online)

<http://www.kkits.or.kr>

# A Design and Implementation of PDIF Converter

Sun-Young Lee<sup>1</sup>, Chung-Ho Hwang<sup>2</sup>, Young-Ho Nam<sup>\*1</sup>

<sup>1</sup>*Department of Computer Science, GyeongSang National University*

<sup>2</sup>*Appia Engineering, Ltd*

### ABSTRACT

The aviation software is developed by dividing the object code part and the parameter part into files. This is because various parameters need to be corrected for optimized flight control, so it is possible to correct only by modifying the parameter file without changing the configuration of the object code. As a result, if the changed parameter is within the valid range proved by the test for the code, the test cost can be reduced because the test for the code verification is unnecessary. These parameters are referred to as Parameter Data Items(PDI) in DO-178C, the standard for aviation software conformance. The conversion of this PDI into a binary file is called PDIF. In this paper, We design and implement a PDIF converter that generates an Excel sheet gain set with various flight scenarios for flight control law data and converts it into a PDIF set. The implemented PDIF converter can perform flight test without changing its configuration when a test is applied to the dynamic environment of Hardware In the Loop Simulation(HILS) for flight safety. The use of the Converter improves test convenience and saves test time as it enables continuous testing of various flight scenarios through PDIF set without interruption compared to the conventional sequential test for each single flight scenario. In addition, the (.dat) file can apply to other types of OFP and reduce the impact of OFP modification.

© 2020 KKITS All rights reserved

**KEYWORDS :** Parameter Data Item, Parameter Data Item File, DO-178C, Operational Flight Program, S-Record

**ARTICLE INFO:** Received 3 August 2020, Revised 10 September 2020, Accepted 13 October 2020.

\*Corresponding author is with the Department of Computer Science, Gyeongsang National University, 501 Jinju-daero, Jinju, 52828, KOREA.

E-mail address: [yhnam@gnu.ac.kr](mailto:yhnam@gnu.ac.kr)

## 1. 서론

항공전자 시스템은 비행 중 발생하는 고장 또는 작동 불능에 대한 대처가 용이하지 않고, 사고 발생에 따른 인명 및 물질 피해가 매우 크기 때문에 신뢰성과 안전성을 최우선으로 고려해야 한다. 특히 항공 분야 기술 발전에 따라 관련 소프트웨어의 복잡도가 높아지고 그에 따른 결함 사례 검출이 늘어나면서, 항공전자 시스템 관련 소프트웨어의 신뢰성 및 안전성을 확보하는 연구가 필요하다.

국내외 항공전자 소프트웨어 개발 연구는 소프트웨어 검증 및 인증을 위한 모델링 연구[1, 2], 항공시스템 및 장비인증 소프트웨어 고려사항의 내재화를 위한 비행운용 프로그램(OPF: Operational Flight Program) 측면에서의 연구[3], 모델링을 통해 산출되어지는 제어법칙 자동코드에 대한 DO-331(Model Based Development) 적용 연구[4], 매개변수 데이터 아이템 파일(PDIF: Parameter Data Item File)의 활용 방안 연구[5, 6] 등이 진행되고 있다.

일반적으로 항공전자 소프트웨어는 목적코드 부분과 매개변수 부분을 파일로 분리하여 개발된다. 이러한 이유는 최적화된 비행제어를 위해 다양한 매개변수의 보정이 필요함에 따라, 코드의 형상변경 없이 매개변수 파일의 수정만으로 보정이 가능하기 때문이다[5]. 이 결과 변경된 매개변수가 코드에 대한 시험으로 입증된 유효범위 내에 있다면 코드 입증을 위한 시험이 필요 없어 시험 비용을 줄일 수 있다.

DO-178C[7]에서는 이 매개변수를 매개변수 데이터 아이템(PDI: Parameter Data Item)이라고 하는데, 실행 가능한 목적코드를 변경할 필요 없이 소프트웨어의 거동에 영향을 주며, 개별적인 형상 항목으로 관리되는 데이터 세트로 정의하고 있다. PDI 자체는 사람이 읽을 수 있는 아스키(ASCII)파

일, 엑셀파일 등으로 작성하고, 플래시 메모리에 저장할 수 있는 이진파일로 변환된 것을 PDIF라고 한다.

본 논문에서는 형상변경 없이 OPF를 시험하기 위해 비행제어 데이터에 대한 다양한 시나리오를 가지는 PDIF를 자동으로 생성하는 PDIF 변환기를 설계하고 구현한다. PDIF 변환기는 PDI와 해당 값들로 구성된 단일 비행시나리오를 저장하고 있는 C파일을 파싱하여 엑셀파일에 PDI와 관련 값들을 저장한다. 시험자는 이 시트화된 데이터를 바탕으로 특정 항목의 추가 및 값의 수정을 통하여 다양한 비행시나리오를 생성하여 저장한다. 변환기는 이 엑셀파일로부터 이진형식과 모토로라형식의 PDIF를 생성한다.

이 PDIF에는 다양한 비행시나리오가 저장되어 있어, 비행제어 시뮬레이터를 한 번 구동시킨 후 중단 없이 필요한 모든 비행시나리오를 연속적으로 시험할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 PDIF를 정의하고 설계 시에 고려사항들을 기술한다. 3장에서는 PDIF 메모리맵 설계와 PDIF 변환기의 기능 설계에 대해 설명하고, 4장에서는 PDIF 변환기의 각 기능 구현에 관하여 기술한다. 마지막으로 5장에서는 본 연구의 결과에 대해 기술한다.

## 2. PDIF

DO-178C에서는 최신 소프트웨어 개발 기법에 따라 개발 및 인증을 위한 보충 문서로 DO-331, DO-332, DO-333이 별도로 제정되었다[8-10]. DO-331은 모델 기반 개발 및 검증(Model-Based Development and Verification Supplement to DO-178C)이나 연구, DO-332는 객체지향 기술 및 관련 기술(Object-Oriented Technology and Related Techniques Supplement to DO-178C) 추가 지침이

다. DO-333은 정형 기법이 소프트웨어 생명 주기의 일부로 사용될 때 필요한 추가 지침이다.

PDI는 DO-178C의 개정을 통해 처음으로 도입된 개념으로, ‘타깃 컴퓨터의 처리 유니트에서 직접적으로 사용되는 데이터의 모음’ 이다[11]. PDI는 실행 가능한 목적 코드에 수정을 가하지 않으면서 실행 시 참조되어 소프트웨어의 거동에 영향을 주는 속성을 포함하기 때문에 PDIF를 통해 별도의 형상 항목으로 관리하려는 것이다. PDIF는 별도의 데이터로써 제공되어야 하고, 타깃 시스템에 업로드 된 이후 지워지지 않아야 하므로, 비휘발성 메모리인 ROM 또는 플래시 메모리에 적재되는 것이 적합하다.

PDIF를 비휘발성 메모리에 적재하기 위해서는 순수 이진 데이터로 구성된 데이터 파일 포맷을 이용하거나, 또는 타깃 CPU에서 지원하는 포맷을 사용하는 방법이 있다. 순수 이진 데이터로만 구성된 파일 포맷을 사용할 때, CPU가 데이터를 처리하는 방식이 빅 엔디안(big-endian)을 사용하는지, 리틀 엔디안(little-endian)을 사용하는지를 고려해야 한다. 그리고 이진 파일에는 데이터 업로드를 위한 메모리 주소 정보가 없고 순수 이진 데이터만 존재하므로, 데이터를 적재하는 프로그램에서 데이터 업로드 주소를 고려해야 한다. 타깃 CPU에서 지원하는 파일 형식으로는 인텔의 HEX와 모토로라의 S-record가 있다. 두 형식은 아스키(ASCII)로 이진 데이터를 전달하도록 지원한다. 본 논문에서는 모토로라 Power-PC 계열의 CPU를 위해 S-record 형식[12]을 사용한다. S-record는 레코드 타입, 바이트 수, 주소, 데이터, 체크섬으로 구성된다. 타깃 CPU에서 지원하는 아스키 형식은 순수 이진 데이터와 다르게 주소 데이터를 가지고 있으므로, 데이터를 적재하는 프로그램에서 업로드 메모리 주소 정보를 지니고 있지 않아도 된다.

또한, PDIF는 비행제어법칙 설계[13, 14], 시뮬레

이션 환경구성 연구[15]에서 동역학 환경이 적용된 연구 및 시뮬레이션할 시, PDI 데이터 세트의 적용이 가능하다.

### 3. PDIF 변환기 설계

#### 3.1 프로그램 설계

PDIF 변환 프로그램은 <그림 1>과 같이 GUI CSC(Computer Software Component), Processing CSC, Save CSC로 나뉘어진다.

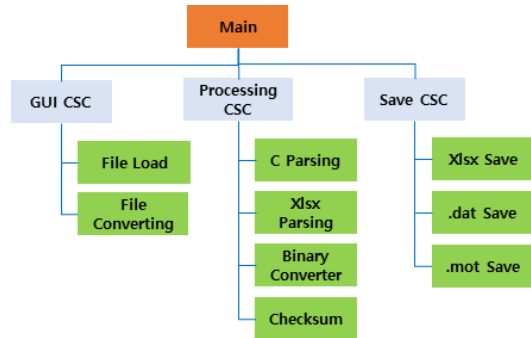


그림 1. PDIF 변환 프로그램 아키텍처  
Figure 1. PDIF Convert Program Architecture

GUI 부분은 소프트웨어 사용자와 인터페이스를 제공하는 모듈이며, Processing 모듈은 각 조건에 맞는 파싱 및 변환 기능을 제공한다. Save 기능은 파싱 및 변환된 데이터를 다양한 형식의 파일로 저장하는 기능을 제공한다.

#### 3.2 PDIF 설계

##### 1) 메모리 맵 설계

<그림 2>는 OFF의 비행 시나리오(Gain Set 1~30)가 입력되는 메모리 영역(Block 01~30)을 도식한 것

이다.

PDIF Area (Flash Memory : 0x10000 ~ 0x2FFFF)			
Memory Block	Address	Size	Data Info
Block 00	0x10000 ~ 0x10003	4096byte	Gain Set Counter
	0x10004 ~ 0x1000B	8192byte	PDIF Project
	0x1000C ~ 0x10013	8192byte	PDIF ID
	0x10014 ~ 0x1001B	8192byte	PDIF Check Sum
	0x1001C ~ 0x10FFF	4068byte	Reserved Memory
Block 01	0x11000 ~ 0x11FFF	4096byte	Gain Set 1
Block 02	0x12000 ~ 0x12FFF	4096byte	Gain Set 2
Block 03	0x13000 ~ 0x13FFF	4096byte	Gain Set 3
Block 04	0x14000 ~ 0x14FFF	4096byte	Gain Set 4
Block 05	0x15000 ~ 0x15FFF	4096byte	Gain Set 5

...			
Block 26	0x2A000 ~ 0x2AFFF	4096byte	Gain Set 26
Block 27	0x2B000 ~ 0x2BFFF	4096byte	Gain Set 27
Block 28	0x2C000 ~ 0x2CFFF	4096byte	Gain Set 28
Block 29	0x2D000 ~ 0x2DFFF	4096byte	Gain Set 29
Block 30	0x2E000 ~ 0x2EFFF	4096byte	Gain Set 30
Block 31	0x2F000 ~ 0x2FFFF	4096byte	Reserved Memory

그림 2. PDIF 메모리맵 설계  
Figure 2. Design of PDIF Memory-map

Block 00번은 시나리오의 정보를 나타내며, Block 01~30은 각 비행 시나리오 데이터를 저장할 수 있는 영역이고, 다음의 형식으로 구성된다.

- Gain Set Counter: 적용된 비행 시나리오(Gain Set)의 개수
- PDIF Project: 수행할 프로젝트 이름
- PDIF ID: 수행할 프로젝트 ID
- PDIF Checksum: 적용된 비행 시나리오 데이터들의 총합

## 2) PDIF를 위한 엑셀 파일 설계

비행 시나리오 데이터는 먼저 Matlab- Simulink를 통해 자동 생성된 global\_Param.c에서 파싱되어 엑셀 파일로 저장되며, 총 31개의 엑셀시트로 구성된다.

Data 시트는 <그림 3>과 같이 비행제어법칙(aviation control law) 데이터의 이름과 데이터의 유효범위(최솟값/최댓값) 필드로 구성되어 있다.

<그림 4>의 Set 1 ~ Set 30에는 30개의 비행 시나리오를 구성할 비행제어법칙 데이터 값이 저장된다. 각 Set에는 500개의 비행제어법칙 데이터를 저장할 수 있으며, 각 데이터들은 최대 17개의 배열 크기를 지원한다. 또한 시험자는 각 Set의 데이터에 대해 복사 및 수정을 통해 비행 시나리오 데이터 값을 설정할 수 있다.

	NAME	MIN	MAX
2	K_col	0	1
3	K_lat	0	1
4	K_lat_trim	0	1
5	K_lon	0	1
6	K_lon_trim	0	1
7	K_ped	0	1
8	K_phi	0	1
9	Kpsi	0	1
10	Ky	0	1
11	kzM	0	1
12	kzN	0	1
13	Kped_col	0	1
14	Kp	0	1
15	Kphi	0	1
16	KphiI	0	1
17	Kpsi	0	10
18	Kpsidot	0	10
19	KpsiI	0	10
20	Kq	0	1
21	Kr	0	1
22	ktheta	0	1
23	kthetaI	0	1
24	Kv	0	1
25	TORQUE	0	150
26	Ta	0	50
27	TaM_double	0	10
28	TaM_square	0	25
29	Tb	0	50
30	TbM_double	0	10
31	TbM_square	0	25
32	Td	0	50
33	TdM_double	0	10
34	TdM_square	0	25
35	TrimOff_coll	0	8
36	TrimOn_coll	0	8
37	TrimOff_pitch	0	8
38	TrimOn_pitch	0	8
39	TrimOff_roll	0	8
40	TrimOn_roll	0	8
41	TrimOff_yaw	0	8
42	TrimOn_yaw	0	8
43	Ts_APCmd	0	10
44	dVt_torque	0	100
45	freq_bilinear_sump_pitch	0	10
46	freq_bilinear_sump_roll	0	10
47	freq_sump_AP	0	10
48	freq_linear_sump_dias	0	10
49	lon_sas_lim	0	1
50	lat_sas_lim	0	1
51	ped_sas_lim	0	1
52	ground_gain_ratio	0	1
53	ga_sas_threshold	0	5
54	ga_vs_threshold	0	5
55	ga_wf_threshold	0	5
56	init_time_sec	0	10
57	upper_Ku	0	1
58	upper_Kui	0	1
59	upper_Kv	0	1
60	upper_Kvi	0	1
61	upper_Kvs_ga	0	1
62	limL_alt_ft	-10000	0
63	limL_PitchCmd	-1	0

그림 3. 비행제어법칙 데이터  
Figure 3. Data of Aviation Control Law

그림 4. 비행 시나리오 데이터 저장 시트  
Figure 4. Flight Scenario Data Storage Sheet

비행 시나리오 데이터 형식은 다음과 같다.

- G\_Set: Gain Set Count
- PJ: PDFIF Project Version
- ID: PDFIF\_ID
- Type: 데이터 타입 종류
- Format: 데이터 형식
- Form: 데이터 형태(단일 데이터: Single, 배열: Array)
- Size: 데이터 크기(단일 데이터면 1, 배열이면 배열크기)
- Name: 데이터 이름
- 0~16: double 형식의 데이터 값

### 3.3 파싱 기능 설계

파싱은 <그림 5>에서 “globalParam.c” 인 제어 법칙 데이터만을 추출하는 간단한 컴파일러 기능을 제공한다.

globalParam.c는 Matlab-Simulink를 통해 자동 생성된 제어법칙코드로 고/저속 유지를 위한 레이트 임계값, 테일로터 블레이드 각도에 대한 페달명령 등으로 이루어진 코드로 비행제어에 참조되는 데

이터들이다.

```

/*
 * File: globalParam.c
 *
 * Code generated for Simulink model 'Model'.
 *
 * Model version      : 1.1762
 * Simulink Coder version : 8.11 (R2016b) 25-Aug-2016
 * C/C++ source code generated on : Mar 04 14:57:32 2020
 *
 * Target selection: ert-tic
 * Embedded hardware selection: Freescale->32-bit PowerPC
 * Code generation objectives:
 *   1. MISRA C:2012 guidelines
 *   2. Traceability
 * Validation result: Not run
 */

#include "rtwtypes.h"
#include "Model_types.h"

/* Exported data definition */

/* Definition for custom storage class: ExportToFile */
real_T AIRSPEED[3] = { 0.0, 20.576, 82.304 };

real_T K_col = 0.113;
real_T K_lat = 0.12;
real_T K_lat_trim = 0.069813;
real_T K_lon = 0.2;
real_T K_lon_trim = 0.034907;
real_T K_ped = 0.5;
real_T KaM = 0.2;
real_T Kay[3] = { 0.0, 0.0, 0.0 };

real_T KbM = 0.4;
real_T KdM = 0.2;
real_T Kp[3] = { 0.05, 0.03, 0.02 };
    
```

그림 5. globalParam.c 파일의 구성  
Figure 5. Organization of golbalParam.c file

globalParam.c 파일은 주석문과, C/C++ 파일의 헤더파일, 제어법칙 데이터로 구성되어 있다. 이중 비행제어법칙 데이터만을 추출하기 위해 변수 Type인 real\_T가 읽히면 변수 영역 검사를 통해 Type(real\_T)과 Format(double형), Form(Sigle, Array)을 체크 후 “=” 이후 데이터는 비행제어 법칙 데이터이므로 스트링형 배열에 모두 저장한다. 저장된 스트링형 배열을 기반으로 PDFIF의 시트(Set1)에 파싱된 비행제어법칙 데이터들을 저장한다.

### 3.4 (.dat) 및 (.mot) 변환 기능 설계

파싱을 통해 PDFIF에 기록된 비행제어법칙 데이터를 읽어서 이진 데이터 형식으로 변환(.dat)한다.

변환 시에 <그림3>의 PDIF의 데이터 시트에서 변환되는 변수의 최솟값/최댓값 범위를 검증한다. 데이터 범위를 벗어나면 오류가 있는 엑셀 시트의 해당 위치를 표시한 후 변환을 중지한다.

파싱을 통해 PDIF에 기록된 비행제어법칙 데이터는 타킷 컴퓨터의 처리 유닛에 맞도록 변환되어야 한다. 모토로라 PowerPC 계열의 CPU에서는 아스키 형식의 텍스트 데이터를 요구한다. 이를 위해 .dat 파일을 아스키 16진수 텍스트로 구성된 S-record 형식으로 변환하도록 설계하였다.

### 3.5 메인화면 구성

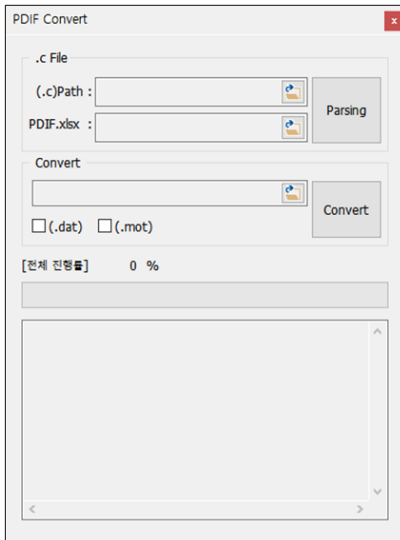


그림 6. PDF 변환기 UI  
Figure 6. UI of PDF Converter

<그림 6>은 PDF 변환기의 UI를 설계한 것이며, 각 필드와 버튼의 기능은 다음과 같다.

- (.c)Path: “globalParam.c“ 파일을 불러오는 버튼
- PDF.xlsx: PDIF 파일을 불러오는 버튼
- Parsing: 파싱을 수행하는 버튼
- 변환될 데이터 파일(PDIF.xlsx)을 불러오는 버튼

- 변환할 데이터 (.dat), (.mot)를 선택하기 위한 체크박스
- Convert: (.dat), (.mot) 파일을 생성하기 위한 버튼
- 파일 생성 진행률을 표시하는 진행 바
- 파일 생성 관련 정보를 나타내는 출력란

## 4. PDIF 변환기 구현

### 4.1. 파싱

PDIF 변환기는 <그림 7>과 같이 입력된 globalParam.c에서 비행제어법칙 데이터를 추출하여 각 데이터 범위를 검증하며 파싱하며, 파싱된 결과를 엑셀파일 형식의 PDIF에 저장한다. 진행률 바를 통해, 전체 진행률이 표시되며, 데이터 변환 및 결과는 데이터 출력란에 표시된다.

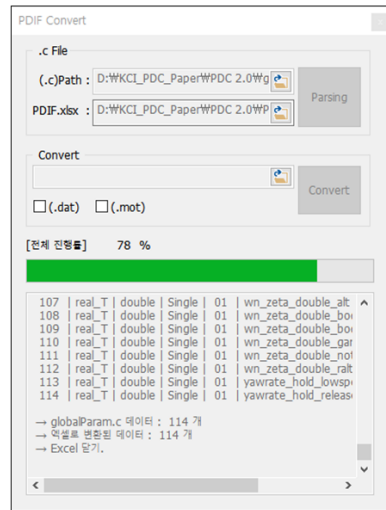


그림 7. 비행제어법칙 데이터 파싱  
Figure 7. Parsing of Aviation Control Low Data

<그림 8>은 globalParam.c에서 파싱된 비행제어법칙 데이터를 엑셀 시트에 저장한 결과이다.

그림 8. 파싱된 비행제어법칙 데이터  
Figure 8. Parsed Flight Control Law Data

#### 4.2. (.dat) 파일 생성

불러오기 버튼을 클릭하여 PDFIF.xlsx를 불러온 후 (.dat) 체크박스를 선택하고 Convert 버튼을 클릭하면, <그림 9>와 같이 PDFIF.dat가 생성된다.

PDFIF.dat에는 <그림 10>과 같이 PDFIF의 Set1에서 Set30 중 데이터 값이 있는 Set의 모든 데이터 값이 이진 값으로 변환되어 저장된다.

생성된 PDFIF.dat의 예는 <그림 11>과 같다.

PDFIF.xlsx	(.dat) data
Gain Set Counter : 1	000000014C43483341
PDFIF_PJ : LCH3AF00	463030
PDFIF_ID : PX01AE00	505830314145303040
PDFIF Check Sum : 3089.79	A823947AE147AB
data	(.dat)
0.0	0000000000000000
20.576	40349374BC6A7EFA
82.304	40549374BC6A7EFA
0.113	3FBCE916872B021

그림 10. PDFIF에서 이진 데이터 생성  
Figure 10. Generating Binary Data from PDFIF

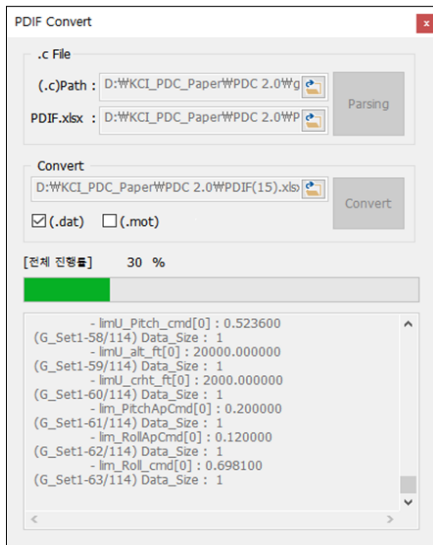


그림 9. (.dat) 파일 생성  
Figure 9. Generating (.dat) file

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00001000	00	00	00	00	00	00	00	40	34	93	74	BC	6A	7E	FA	
00001010	40	54	93	74	BC	6A	7E	FA	3F	BC	ED	91	68	72	B0	21
00001020	3F	BE	B8	51	EB	85	1E	B8	3F	B1	DF	43	C7	D5	ED	07
00001030	3F	C9	99	99	99	99	99	9A	3F	A1	DF	54	8E	CD	8D	BD
00001040	3F	E0	00	00	00	00	00	00	00	3F	C9	99	99	99	99	9A
00001050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00001060	00	00	00	00	00	00	00	00	00	3F	D9	99	99	99	99	9A
00001070	3F	C9	99	99	99	99	99	9A	3F	A9	99	99	99	99	99	9A
00001080	3F	9E	B8	51	EB	85	1E	B8	3F	94	7A	E1	47	AE	14	7B
00001090	3F	DB	AC	71	0C	B2	95	EA	3F	E0	C4	9B	A5	E3	F8	
000010A0	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
000010B0	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
000010C0	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
000010D0	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
000010E0	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
000010F0	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
00001100	3F	D0	74	53	8E	F3	4D	6A	3F	D0	74	53	8E	F3	4D	6A
00001110	3F	D0	74	53	8E	F3	4D	6A	3F	C7	0A	3D	70	A3	D7	0A
00001120	3F	C1	EB	85	1E	B8	51	EC	3F	B9	99	99	99	99	99	9A
00001130	3F	B3	33	33	33	33	33	33	3F	AE	B8	51	EB	85	1E	B8
00001140	3F	A9	99	99	99	99	99	9A	3F	F0	00	00	00	00	00	00
00001150	3F	F0	00	00	00	00	00	00	3F	F0	00	00	00	00	00	00
00001160	3F	F0	00	00	00	00	00	00	00	3F	F0	00	00	00	00	00
00001170	3F	F0	00	00	00	00	00	00	00	3F	F0	00	00	00	00	00
00001180	3F	F0	00	00	00	00	00	00	00	3F	F0	00	00	00	00	00
00001190	3F	C9	99	99	99	99	99	9A	3F	C9	99	99	99	99	99	9A
000011A0	3F	C9	99	99	99	99	99	9A	3F	A9	99	99	99	99	99	9A
000011B0	3F	B9	99	99	99	99	99	9A	3F	D9	99	99	99	99	99	9A
000011C0	3F	D3	33	33	33	33	33	33	3F	D3	33	33	33	33	33	33
000011D0	3F	D3	33	33	33	33	33	33	3F	C3	33	33	33	33	33	33
000011E0	3F	BE	B8	51	EB	85	1E	B8	3F	B9	99	99	99	99	99	9A
000011F0	3F	A9	99	99	99	99	99	9A	3F	C9	99	99	99	99	99	9A

그림 11. 생성된 PDFIF.dat  
Figure 11. Generated PDFIF.dat

PDFIF 변환 프로그램은 엑셀파일 형식의 PDFIF에 저장되어 있는 데이터를 이진 데이터로 변환한다.





- (ICICT), Kahului, HI, USA, pp. 285-291, 2019.
- [3] N. Park, *A development of OFP auto code generating tool for air vehicle built-In test*, Proceeding of the 2018 KSAS Fall Conference, pp. 609-613. 2018.
- [4] S. Ahn, *Secure methodology of the autocode integrity for the helicopter fly-by-wire control law using formal verification tool*, Proceeding of the 2014 KSAS Fall Conference, pp. 398-405. 2014.
- [5] S. K. Paek, *Embedded software design using parameter data item*, Proceeding of the 2016 KSAS Fall Conference, pp. 553-554, 2016.
- [6] S. Y. LEE, *Proposed PDIF generation program for flight testing in various scenarios*, Proceeding of the 2019 KSAS Fall Conference, pp. 1403-1404, 2019.
- [7] W. K. Youn, S. B. Hong, K. R. Oh, and O. S. Ahn, *Software certification of safety-critical avionic systems: DO-178C and its impacts*, in IEEE Aerospace and Electronic Systems Magazine, Vol. 30, No. 4, pp. 4-13, 2015.
- [8] M. Hochstrasser, S. P. Schatz, K. Nürnberger, M. Hornauer, S. Myschik, and F. Holzapfel. *Aspects of a consistent modeling environment for DO-331 design model development of flight control algorithms*, in Springer, Cham, Advances in Aerospace Guidance, Navigation and Control, pp. 69-86, 2017.
- [9] M. Hochstrasser, S. Myschik, and F. Holzapfel, *Application of a process-oriented build tool for flight controller development along a DO-178C/DO-331 process*, in Springer, Cham, Model-Driven Engineering and Software Development. Vol 991, pp. 380-405, 2019.
- [10] A. Paz, and G. El Boussaidi, *On the exploration of model-based support for DO-178C-compliant avionics software development and certification*, 2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Ottawa, ON, pp. 229-236, 2016.
- [11] H. M. Joe, *A study on means of compliance of airborne software per DO-178C*, KCA Vol 64, pp. 61-79, 2016.
- [12] WIKI,. <https://ko.wikipedia.org/wiki/SREC>, Aug. 2020.
- [13] N. E. Rodnishchev, L. G. Romanenko, and K. G. Denisov, *To estimation of control law parameters for the lateral motion of aircraft taking into account wind disturbances*, Russian Aeronaut. 58, pp. 407-412, 2015.
- [14] J. Holsten, and D. Moormann, *Flight control law design criteria for the transition phase for a tiltwing aircraft using multi-objective parameter synthesis*, CEAS Aeronaut J 6, pp. 17-30, 2015.
- [15] H. J. Choi, *Design of an integrated flight control model for light civil helicopter and configuration of the simulation environment with the dynamics model*, Proceeding of the 2019 KSAS Fall Conference, pp. 1040-1041, 2019.

---

## PDIF 변환기의 설계 및 구현

이순영<sup>1</sup>, 황충호<sup>2</sup>, 남영호<sup>3</sup>

<sup>1</sup>경상대학교 컴퓨터과학과 박사과정

<sup>2</sup>아피아 엔지니어링 선임연구원

<sup>3</sup>경상대학교 컴퓨터과학과 교수

---

## 요 약

항공용 소프트웨어는 목적코드 부분과 매개변수 부분을 파일로 분리하여 개발된다. 이러한 이유는 최적화된 비행제어를 위해 다양한 매개변수의 보정이 필요함에 따라, 목적코드의 형상변경 없이 매개변수 파일의 수정만으로 보정이 가능하기 때문이다. 이 결과 변경된 매개변수가 코드에 대한 시험으로 입증된 유효범위 내에 있다면 코드 입증을 위한 시험이 필요 없어 시험 비용을 줄일 수 있다. 이러한 매개변수를 항공용 소프트웨어 적합성 입증 표준인 DO-178C에서는 매개변수 데이터 아이템(PDI)이라고 한다. 이 PDI를 이진파일로 변환한 것을 PDIF라고 한다. 본 논문에서는 비행제어법칙 데이터에 대한 다양한 비행 시나리오를 가지는 엑셀시트의 Gain Set를 생성하고 PDIF Set으로 변환하는 PDIF 변환기를 설계 및 구현한다. 구현한 PDIF 변환기는 비행안전성을 위한 HILS(Hardware In the Loop Simulation)의 동역학 환경이 적용된 시험을 할 때, 형상변경 없이 비행시험을 수행할 수 있다. 본 연구에서 개발한 변환기의 이용은 기존 단일 비행시나리오별 순차시험에 비해 PDIF Set을 통한 다양한 비행 시나리오 시험을 중단 없이 연속적으로 수행할 수 있어 시험의 편리성이 향상되었고 시험 시간을 절약할 수 있다. 또한 (.dat) 파일을 통해 다른 형식의 OFP에 적용 및 OFP 수정에 대한 영향성을 줄일 수 있다.



**Chung-Ho Hwang** received the M.S. degree in the Department of Convergence Software from Gyeongsang National University in 2015. His research interests include

Aviation Software.

*E-mail address:* qweo89@naver.com



**Young-Ho Nam** received the Ph.D. degree in the Department of Computer Engineering from the Chung-Ang University in 1994. He has been a

professor in the Department of Computer Science at Gyeongsang University since 1996. His current research interests include embedded system, IoT and computer science education. He is a member of the KKITS.

*E-mail address:* yhnam@gnu.ac.kr



**Sun-Young Lee** received the M.S. degree in the Department of Information Science from Gyeongsang National University in 2016.

His research interests include

Aviation Software. He is a member of the KSAS.

*E-mail address:* astesia@gnu.ac.kr