

RS422 버스를 이용한 저속통제기 UFC/DC 데이터 통신 기법 개발

Development of UFC/DC Data Communication method for XKO-1 using RS-422 Bus

양 승 열* 김 영 택**
 Yang, Seung-Yul Kim, Young-Taek

ABSTRACT

ASC(Avionics System Computer) was developed to control weapon delivery and navigation sensors, and to perform man-machine interface with pilots for XKO-1 aircraft. The data communications between ASC and UFC(Up Front Controller), DC(Data Concentrator) were implemented by RS422 serial data bus. Also, SCIL(Standard Computer Interface Library) was designed to facilitate control and management of the computer hardware resources and is embedded in the ASC. These structures have a merit of noise immunity and a reduction of wire harness for signal lines, and enable OFP(Operational Flight Program) programmers to use the SCIL easily without knowing hardware details. Manufactured system was on installed on XKO-1, and performed for BIT(Built In Test) and interface test with UFC and DC. The test results show that it meets the system requirements.

주요기술용어 : XKO-1(저속통제기), ASC(항공전자시스템 컴퓨터), UFC(전방상향 제어패널), DC(Data Concentrator), SCIL(연동라이브러리), OFP(항공전자 운용소프트웨어)

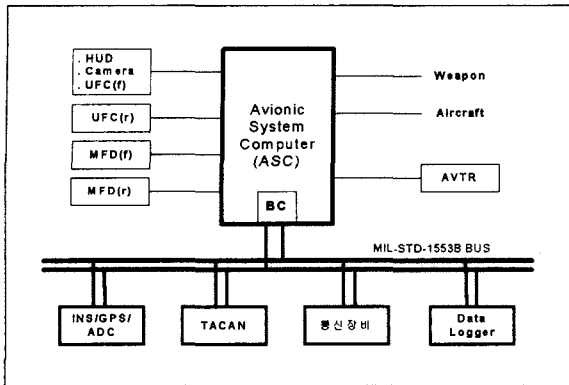
1. 서 론

군용 항공기의 비행, 항법, 통신 및 무장제어 요구 성능을 충족시키기 위해 탑재되는 항공전자 시스템은 조종사의 임무부하를 최소화 하여 효율적으로 시스템을 운용할 수 있는 통합 정보전시 및 제어능력이 필

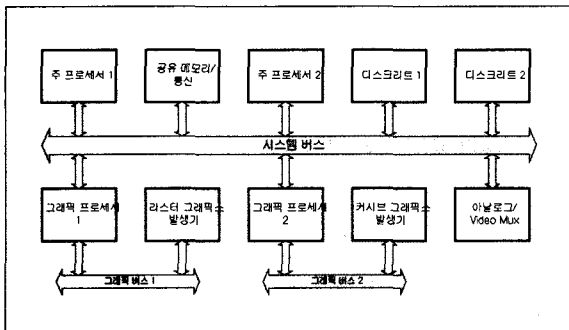
수적이다. 본 논문은 이를위해 개발된 항공전자 시스템 컴퓨터(ASC, Avionics System Computer, 이하 ASC)에서 RS422 버스를 통해 연동되는 통신모듈의 하드웨어와 소프트웨어 설계내용에 대해 기술하였다. ASC는 항공전자 무장제어 시스템을 중앙에서 통합하고 제어하는 핵심 구성품으로서 군용 항공전자 시스템의 간략화된 구성도는 그림 1과 같다.

저속통제기는 현재 양산되고 있는 기본훈련기의 성능향상을 위해 개발되는 항공기로서, 조종사가 시스템

* 국방과학연구소 연구원
 ** 국방과학연구소 선임연구원



[그림 1] 군용 항공전자 시스템 개략 구성도



[그림 2] ASC 내부 전자회로 구성

제어 및 데이터 입력을 위해 전/후방석에 전방상향 제어 패널(UFC, Up Front Controller)이 장착되며, 항공기 heading 및 경로를 입력하는 베젤 제어 패널(BCP, Bezel Control Panel)과 연결되어 있는 DC(Data Concentrator)가 항전베이에 장착된다. 해외 구매품인 UFC, DC는 RS422 데이터 버스 통신기능을 제공하므로 이 장비들과 ASC 간의 통신을 위하여 ASC 내에 직렬통신제어기와 GAL 소자를 이용한 통신보드를 설계, 제작하였다.^[1]

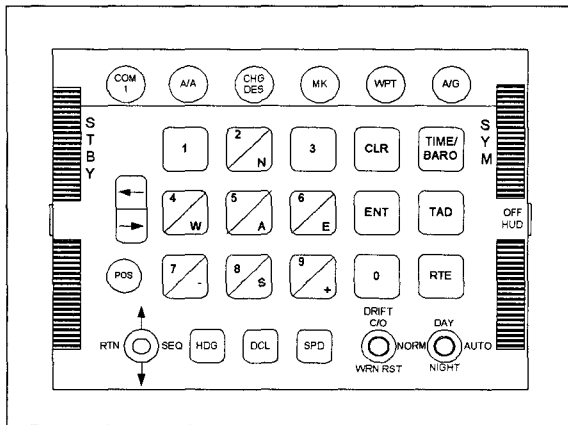
본 논문에서는 상이한 통신 프로토콜을 갖는 장비인 UFC와 DC가 ASC의 응용프로그램인 OFP에서 50Hz 주기의 실시간으로 운용될 수 있도록 최적의 통신을 구현하기 위한 방안을 제시하였으며, 이를 위한

하드웨어 설계와 ASC에 내장되는 시스템 커널 소프트웨어에 대해 기술하였다. ASC는 시제품을 개발하여 실험실에서 전기적 특성시험과 군용 항공기 탑재 환경 적합성 검증을 위한 환경시험 및 전자기 간섭시험을 통해 검증하였으며, 비행시험을 통해 항공기 탑재 비행성능을 확인 하였다.

2. ASC 통신모듈 설계

저속통제기에 장착되는 ASC는 4개의 프로세서 모듈을 포함한 10개의 전자회로 모듈을 한개의 LRU에 통합 구현함으로써 탑재공간 및 중량 감소는 물론 항공전자 및 무장제어 요구성능도 만족시켰다. 그림 2는 ASC 내부 전자회로 구성을 나타낸다. 프로세서는 인텔사의 32비트 RISC 프로세서인 80960 칩을 활용하여 프로세서 모듈을 설계하였다. 따라서 상용 실시간 운영체제를 사용하는 대신 하드웨어를 구동하기 위한 커널 소프트웨어 프로그램을 개발 활용함으로써 응용 소프트웨어의 하드웨어에 대한 독립성을 최대한 확보하였다. UFC/DC와 인터페이스를 담당하는 통신 모듈은 Mil-std-1553B와 RS422 및 ARINC-429 등 직렬 데이터통신 제어기들로 구성된다.

ASC의 통신모듈과 RS422 통신을 하는 UFC는 시스템 모드, 기수방위 및 비행속도 형식, 전방상향 시현기(HUD, Head Up Display) 포맷 레벨 선택과 시스템 관련 데이터를 입력하는 기능을 제공한다. 그림 3은 저속통제기 전/후방석에 탑재되는 UFC를 나타낸 것으로 UFC 내부에는 마이크로 컨트롤러가 내장되어 있어 자체점검을 수행하고 조종사에 의해 눌러진 키 코드가 RS422 버스를 통해 ASC로 전송된다. UFC의 기능 선택 및 데이터 입력은 전/후방석 어디서나



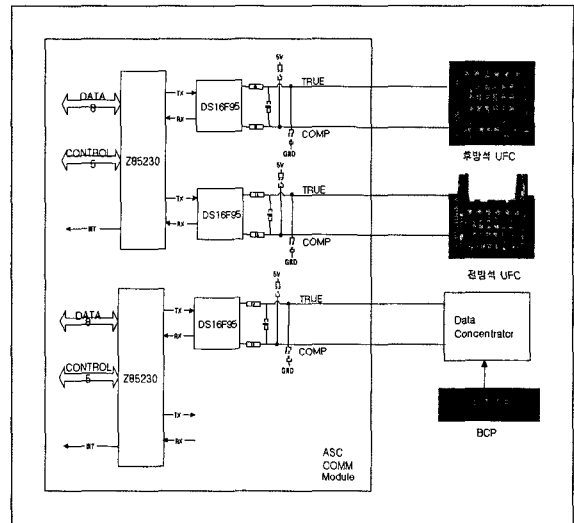
[그림 3] Up Front Controller(UFC)

가능하나, 하나의 UFC만 존재하는 것처럼 입력된 정보가 처리된다.

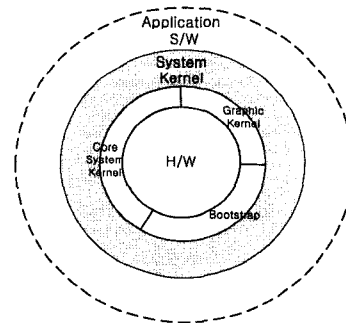
DC(Data Concentrator)는 전자식 계기인 EFI/EEI(Electronic Flight Instrument/Electronic Engine Instrument)에 전시를 위한 데이터 처리기능을 주로 수행하며, 조종석의 베젤제어패널(BCP, Bezel Control Panel)을 통해 입력되는 항공기 heading 및 경로 등에 대한 정보가 DC로 전송되고 이 정보는 다시 RS422 버스를 통해 EFI와 ASC로 제공된다. ASC와 UFC, DC의 인터페이스는 RS422 반이중(Half-Duplex) 방식으로 이루어 지도록 설계 하였으며, 그림 4는 ASC 통신모듈과 전/후방석 UFC, DC와의 인터페이스 블록도를 나타낸 것이다.

3. RS422 커널 소프트웨어

ASC는 복잡 다양한 기능을 수행하는 임무장비로서 각종 탑재장비들과 연동되므로 방대한 응용 S/W의 효율적인 개발환경 제공을 위해 하드웨어의 상세 설계 및 동작에 관한 내용을 숨길(Shield)수 있도록



[그림 4] UFC/DC 인터페이스 회로도



[그림 5] ASC 계층적 구조

H/W와 응용 S/W 간의 인터페이스를 담당할 하위 레벨의 S/W가 필요하다. 따라서 ASC의 H/W 기능 모듈 단위로 구조화된 커널 S/W를 개발하여 ASC의 모든 H/W 동작을 제어함으로써 H/W에 대한 자세한 지식이 없이 응용 S/W 개발이 가능하며, 시스템 형상변경 등에도 쉽게 대처할 수 있도록 하였다.

시스템 커널 소프트웨어는 ASC의 각 H/W에서 독립적인 패키지로 구성하여 응용 S/W 개발자가 참조하도록 각 패키지 Specification에 해당 함수 및 변수들을 정의하고 상세 내용은 패키지 Body에 숨김으로

써 H/W와 응용 S/W 간의 독립성을 최대한 확보하였다. 그림 5는 ASC의 하드웨어와 소프트웨어의 계층적 구조를 나타낸다.^[2]

ASC의 통신모듈과 관련되는 커널 소프트웨어로는 RS-232, RS-422, MIL-STD-1553 B, ARINC-429 등의 직렬 통신을 위한 커널 패키지가 있으며 다양한 외부 연동장비와 데이터 통신이 가능하도록 하였다. 또한 RS-422 직렬 통신 장비들의 통신 프로토콜에 따라 인터럽트 및 폴링 방식으로 제어 루틴을 설계하였다.

3.1 UFC 인터페이스 패키지

전/후방식 UFC와 통신이 가능하도록 하기 위한 UFC 인터페이스 패키지는 반이중(half duplex) 비동기 프로토콜방식을 사용하며, 전송속도는 19,200 bps 이고 1 start 비트, 8 데이터 비트, odd 패리티 비트 그리고 1 stop 비트를 사용한다.

전체적인 통신 알고리즘은 ASC에서 UFC로 먼저 15 바이트의 control block 데이터를 전송한 후 (master), UFC에서는 ASC로 7 바이트의 status block 데이터를 전송하는(slave) 형태로 이루어진다. UFC로 전송되어야 하는 15바이트 control block 데이터 포맷과 UFC로부터 수신되는 7 바이트 status block 데이터 포맷과 전송순서는 표 1, 2와 같다.^[3]

UFC 프로토콜은 50Hz 주기로 운용되는 OFP에서 실시간으로 운용되어야 하는데, 구매한 장비인 UFC의 요구조건을 충족시키기 위해서 15바이트의 송신과 7 바이트의 수신을 폴링방식으로 처리할 경우 ASC의 응용프로그램인 OFP가 50Hz 루틴에서 꼭 필요한 다른 소프트웨어를 수행할 없게 된다. 그러므로 UFC와의 통신에서는 인터럽트 방식으로 통신 하도록 설계하였다.

[표 1] UFC control block 포맷

Name	전송 순서	BYTE FORMAT							
		D7	D6	D5	D4	D3	D2	D1	D0
command	01	UFC LOOP BACK			0	0	0	0	UFC MODE
spare 1	02	0	0	0	0	0	0	0	0
spare 2	03	0	0	0	0	0	0	0	0
spare 3	04	0	0	0	0	0	0	0	0
spare 4	05	0	0	0	0	0	0	0	0
spare 5	06	0	0	0	0	0	0	0	0
spare 6	07	0	0	0	0	0	0	0	0
spare 7	08	0	0	0	0	0	0	0	0
spare 8	09	0	0	0	0	0	0	0	0
spare 9	10	0	0	0	0	0	0	0	0
spare 10	11	0	0	0	0	0	0	0	0
spare 11	12	0	0	0	0	0	0	0	0
spare 12	13	0	0	0	0	0	0	0	0
spare 13	14	0	0	0	0	0	0	0	0
checksum	15	ASC CHECKSUM							

[표 2] UFC status block 포맷

Name	전송 순서	BYTE FORMAT							
		D7	D6	D5	D4	D3	D2	D1	D0
UFC STATUS	01	UFC LOOP BACK			UFC NOR-DCO		UFC ADN		UFC MODE
UFC KEYBOARD	02	0	0	0	UFC-KEY				
UFC FAILURE 1	03	0	0	μ P PSW	μ P ACCUM	μ P STACK	μ P INT RAM	μ P REGS	μ P ALU
UFC FAILURE 2	04	0	0	UFC VDD	UFC KBD	UFC PON RAM	UFC PER RAM	UFC ROM	0
SPARE	05	0	0	0	0	0	0	0	0
UFC S/W VER	06	S/W DIG 2				S/W DIG 1			
UFC CHECKSUM	07	UFC-CHECKSUM							

인터럽트는 ASC의 주프로세서 모듈의 ASIC, 즉 FPGA에 의해 처리되는 것으로 각 인터럽트는 해당 벡터 번호를 가지며 그 인터럽트 레벨을 프로그램 할 수 있다. 전원 공급시 모든 인터럽트는 레벨 0으로 세트되고 레벨 0은 인터럽트를 disable 시킨다. 50Hz, 100Hz 및 Timer 인터럽트는 ASIC내에서 발생되나, 외부 인터럽트들은 ASIC으로 연결되어 ASIC 내부의 인터럽트 제어기에 의해 처리되도록 설계 하였다.

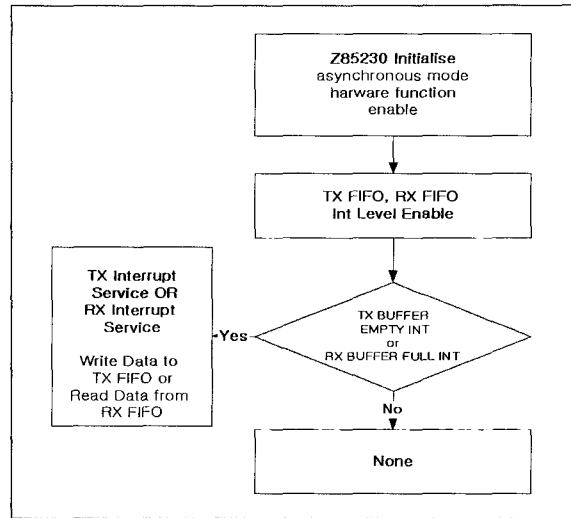
ASC에서 UFC로 전송되는 15 바이트 control block 데이터는 전송되는 바이트 사이에 idle time 없이 전송이 이루어 질 경우, UFC에서 overrun이 발생할 수 있고 이 경우에는 ASC와 UFC 사이의 통신이 잘 이루어지지 않는다. 이것을 방지하기 위하여 직렬 통신링크 상에 flow control을 수행하였는데, 정해진 프로토콜에서 규정한 1 stop 비트를 사용하지 않고 2 stop bit 로해서 전송하여 수신측인 UFC에서 overrun이 발생하지 않도록 하였다.

인터럽트 방식으로 통신이 이루어 지도록 하기 위하여 직렬통신제어기를 인터럽트 I/O 운용모드로 설정하여 사용하였으며, 다음과 같은 조건에서 인터럽트가 발생하도록 하였다.

- Transmitter INT 발생 : 직렬통신제어기의 4 byte TX FIFO가 empty 일 경우
- Receiver INT 발생 : 직렬통신제어기의 RX FIFO에 4 byte가 수신되었을 경우

그림 6은 ASC와 UFC와의 통신에서 동작하고 있는 인터럽트 서비스 운용상태도를 나타내고 있다. 먼저 직렬통신 제어기를 비동기 방식으로 초기화하고 난 후 인터럽트 운용모드로 설정하였다.

다음에 각각의 인터럽트 소스에 따라 4바이트씩 데이터를 TX FIFO에 write 하거나 RX FIFO로부터 4 바이트씩 데이터를 read 하는 인터럽트 서비스 루틴



[그림 6] 인터럽트 방식을 사용한 UFC와 통신

(ISR)을 수행하여 UFC와 통신을 수행하였다.

3.2 DC 인터페이스 패키지

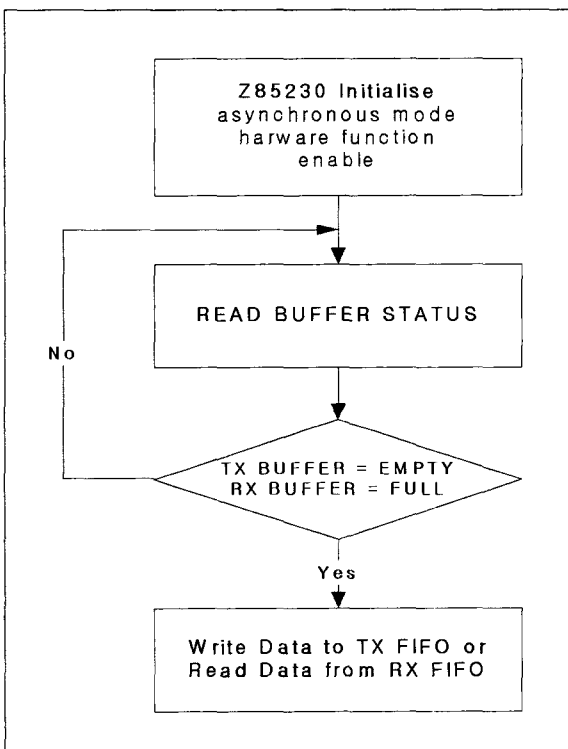
DC와 RS422 통신을 수행하기 위한 DC 인터페이스 패키지는 비동기 프로토콜을 사용하며, 전송속도는 9,600 bps이고 1 start 비트, 8 데이터 비트, no parity 비트 그리고 1 stop 비트를 사용한다.^[4] BCP에서 입력된 헤딩 및 경로설정 데이터는 DC를 통하여 ASC로 전송되고 이 데이터에는 Synchro, CRS counter, HDG counter, checksum, EOT 데이터가 포함되어 있다. DC에서 RS422 데이터 버스를 통하여 ASC로 전송되어 오는 하나의 프레임 형태는 표 3과 같다.

[표 3] DC RS422 전송형태

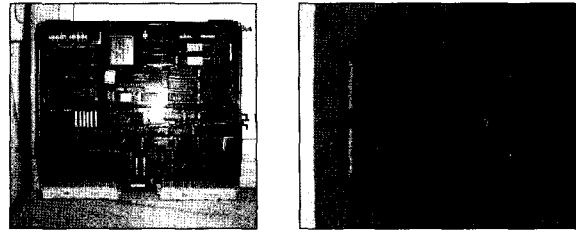
Frame Word Value	0x42	0x00 to 0xFF	0x00 to 0xFF	0x00 to 0xFF	0x0D
Description	SYN-CHRO	CRS counter	HDG counter	Checksum	EOT

여기에서 CRS 및 HDG counter 단위는 1 degree increment로 정의되고 새로운 데이터 프레임은 매 50ms마다 DC에서 전송되어 온다.

ASC의 통신모듈에서는 DC와 통신하기 위하여 폴링방식의 I/O 운용을 사용하는데, 폴링방식은 CPU 제어에 의한 방식으로 직렬통신제어기의 송수신 버퍼에 데이터를 write 하거나 read 할 때 각각의 송수신 버퍼의 상태를 먼저 점검한 후에 데이터를 쓰거나 읽는 방식이다. 그림 7은 DC와 RS422 통신에서 사용되는 폴링 방식을 이용한 데이터 흐름도를 나타낸다. 직렬통신 제어기를 DC 프로토콜에 맞게 설정한 후, 수신 버퍼의 상태를 읽고 버퍼가 Full 되어 있는지 점검한 후 버퍼에서 DC에서 주기적으로 전송되는 5 바이트 데이터를 읽어낸다.



[그림 7] 폴링방식을 사용한 DC와의 통신 흐름도



[그림 8] 통신모듈 및 ASC 제작 외형

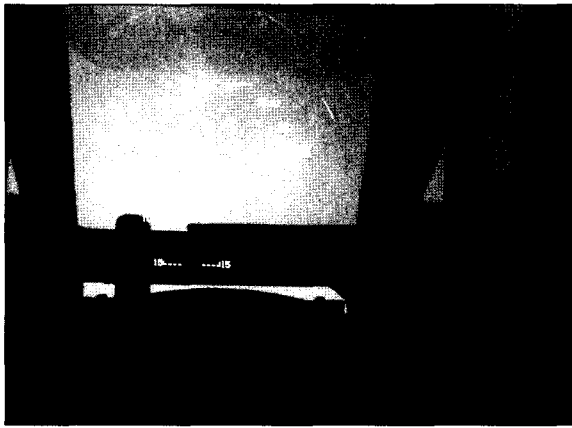
4. 제작 및 시험

RS422 버스를 이용하여 UFC/DC와 통신을 하기위한 ASC의 전체 하드웨어 외형과 통신모듈은 그림 8과 같다.

통신시험 프로그램은 설계된 UFC 및 DC 인터페이스 패키지 커널 소프트웨어를 이용하여 Ada 언어로 코딩하였고, 컴파일 및 링크된 시험 프로그램은 ASC 주 프로세서 보드의 플래쉬 메모리에 이식하였다. 일반 PC에서도 ASC에 응용프로그램을 이식할 수 있도록 별도의 프로그램을 개발하였는데, 이것은 일반 PC에서 동작되는 다운로드 유틸리티와 CPU 보드의 EPROM에 탑재되어 있는 CPU 초기화 및 업로드 프로그램을 이용하였다.^[6] 통신 시험 프로그램은 OFPGA 운용되는 50Hz 주기의 실시간으로 운용되도록 하였으며, 플래쉬 메모리에 저장된 시험 프로그램의 동작상태는 개발된 디버깅 유틸리티를 이용하여 모니터링 하였다.^[7]

4.1 UFC 통신시험

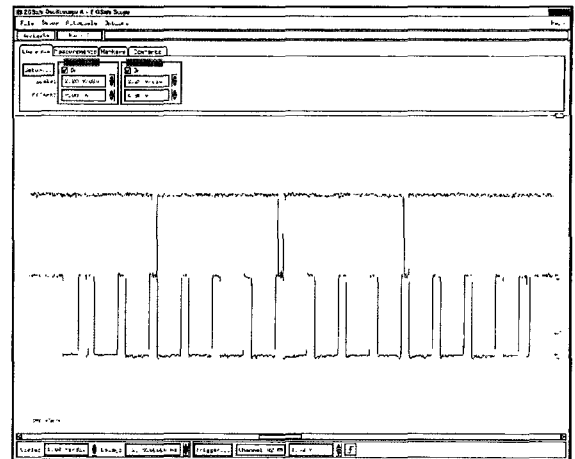
설계 제작된 ASC의 통신모듈과 UFC가 RS422 버스를 이용하여 데이터 통신을 수행하는 과정을 시험 하였다. 먼저 UFC와 통신 시험은 RS422 직렬통신제어기 채널의 자체점검, UFC의 BIT 결과 및 UFC의



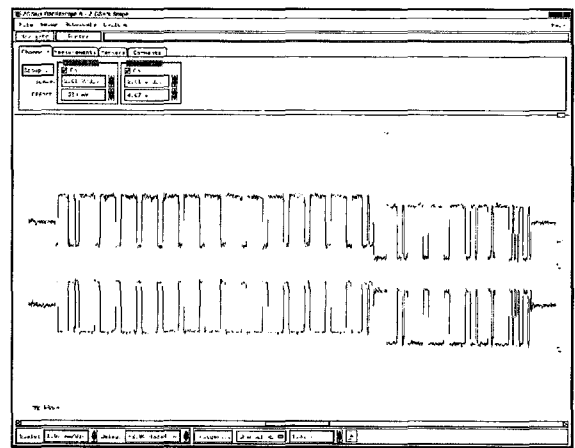
[그림 9] ASC에 의한 HUD 전시화면

키 코드 누름상태를 디버깅 유틸리티를 이용하여 일반 PC 모니터를 통해 전시하는 방법으로 시험 하였다. 그리고 UFC 패키지를 호출하는 응용프로그램인 OFP를 주프로세 보드의 플래쉬롬 메모리 탑재하여 UFC의 키코드를 눌렀을 때 전방상향전시기(HUD)에 전시되는 정보나 심볼의 고정 윈도우 배열중에서 하단에 나타나는 data working line에 그 결과가 정상적으로 전시됨을 확인하였고 시험을 반복 수행하였을 때 오류없이 전시됨을 확인하였다. 그림 9는 UFC 키코드 누름상태를 포함하는 HUD 전시화면을 나타낸다.

또한 로직분석기를 이용하여 전기적 특성을 측정하였으며, 그림 10, 11은 그 결과를 나타낸다. 그림 10은 UFC의 프로토콜에 따라 키코드 누름상태를 ASC에서 수신하기 위해 먼저 ASC가 UFC로 전송하는 15바이트의 control block을 나타내는 것이다. 그림 10의 파형에서 위 부분은 설계된 통신보드의 하드웨어와 UFC 커널의 인터럽트 처리루틴의 실행상태를 표시하며, 아랫부분은 3번의 송신 인터럽트 처리루틴이 실행될 때 ASC에서 UFC로 4바이트씩 데이터가 전송되고 있음을 나타낸다.



[그림 10] ASC에서 UFC로 전송되는 신호 파형

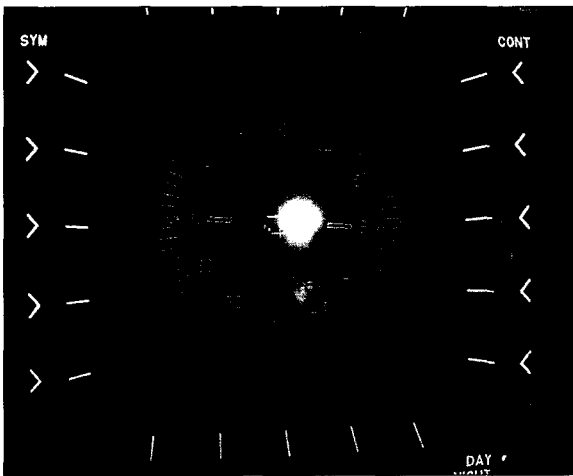


[그림 11] UFC로부터 송수신되는 신호파형

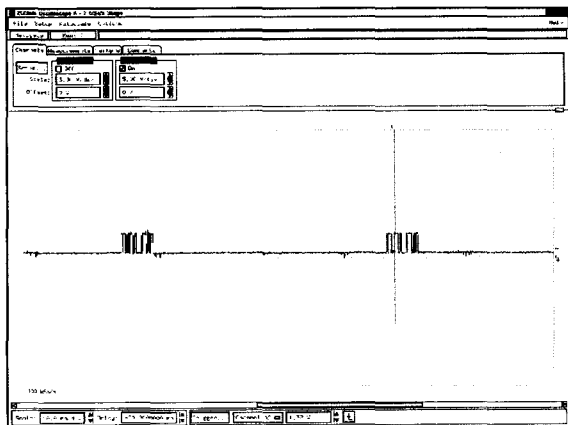
그림 11은 RS422 버스의 differential voltage를 나타내는 것으로 먼저 UFC에서 ASC에서 전송되어온 15바이트를 모두 수신한 후(그림의 왼쪽부분)에 UFC의 키코드 누름 상태를 ASC로 전송하는 7바이트 status block이 전송되는 파형(그림의 오른쪽 부분)을 측정한 결과를 나타낸다. ASC에서 UFC로부터의 7바이트 수신은 2번의 수신 인터럽트 처리루틴으로 동작되며, 통신모듈에서 7바이트의 데이터가 수신되는 것을 보여준다.

4.2 DC 통신시험

설계된 ASC 통신모듈과 DC와의 통신시험은 UFC 통신시험과 마찬가지로 방법으로 수행하였다. 시험프로그램을 작성한 후, DC 인터페이스 패키지를 호출하는 응용프로그램인 OFP를 주프로세 보드의 플래쉬 롬 메모리 탑재하여 BCP의 경로 및 헤딩 정보가 MFD (Multi Function Display)의 오른쪽 상단에 변화된 값이 전시되는지 반복 확인하였다. 그림 12는 헤딩과 경로 데이터를 포함하는 MFD 전시화면을 나타낸다.



[그림 12] ASC에 의한 MFD 전시정보



[그림 13] DC로부터 전송되는 파형

또한 로직분석기를 이용하여 전기적 특성 시험을 수행하였으며, 그 결과는 그림 13과 같다.

그림 13은 DC에서 ASC로 50ms 주기로 전송되는 5 바이트의 신호파형 측정결과로, 신호파형으로부터 폴링방식의 수신루틴이 정상적으로 처리되고 있음을 보여준다.

5. 결 론

본 논문은 저속통제기 항공전자계통을 통합, 제어하기 위한 항공전자시스템 컴퓨터 개발의 일부로서, 같은 RS422 통신방식을 제공하지만 데이터 전송속도 및 인터페이스 방식이 상이한 UFC, DC와 50Hz 주기의 실시간으로 운용되고 있는 OFP가 내장되어 있는 ASC 사이에 최적의 통신을 구현하기 위한 방안을 기술하였다.

또한 통신모듈 하드웨어와 UFC 및 DC 인터페이스 패키지 커널 소프트웨어를 OFP 개발자에게 제공함으로써 OFP 개발자는 ASC 하드웨어에 대한 자세한 지식 없이도 설계자료의 입출력 인터페이스만을 참조하여 효율적으로 OFP 설계를 가능하도록 하였다. 설계된 하드웨어와 소프트웨어는 자체점검 및 UFC, DC와 ASC의 연동시험과 항공기 비행시험을 수행하였으며 시험결과 정상적으로 통신이 이루어지는 것을 확인할 수 있었다.

참 고 문 헌

- [1] 이재역 외 5인, 항공전자 시스템 컴퓨터 하드웨어 설계, 국방과학연구소 연구보고서, 2001.

- [2] 이재억 외 2인, 항공전자 시스템 컴퓨터 연동 라이브러리(SCIL) 설계, 국방과학연구소 연구보고서, 2001.
- [3] UFCP Communication Protocol, Elop electrooptics industries LTD., 2000.
- [4] IRS of system CSCI for Data Concentrator, Sextant, 2000.
- [5] 양승열 외 1인, 항공전자 시스템 컴퓨터 CPU 초기화 및 업로드/다운로드 프로그램 설계, 국방과학연구소 연구보고서, 2002.