KTX 운전자지원 단말기장치 통신프로토콜에 관한 연구

A Study on communication Protocol of KTX TECA(Terminal Cabin)

정성윤* 박신호* 김형인* 김치태*
Jung, Sung-youn Park, Shin-Ho Kim, Hyeoung-In Kim, Chi Tae
강기석** 이병원** 이동수*** 정도원*
Kang, Ki-Sok Lee, Byung-Won Lee, Dong-Soo Jung, Do-Won

ABSTRACT

TECA (Terminal Cabin), the terminal device which offers the information to the driver about the condition of High Speed Train(KTX), analyzes the system of train and plays a role of corrective maintenance guideline to a driver when a breakdown occurs. It also supports the driver to decide the situation of the train by offering a necessary information, in communicating with MPU(Main Processor Unit).

For Localization of TECA, it is necessary to analyze communication protocol between TECA and MPU. As a part of analysis, communication protocol between OBCS(on-board computer system) and add-on devices has analyzed for the first step. Also with protocol analyzer appropriate method was offered for TECA and MPU protocol analysis.

Finally, by appling proposed method to the drivers consol suitability was verified

---

1. 서론

2005년 국가교통핵심기술개발사업의 신규과제로 “경부고속열차 차량컴퓨터제어장치개발”과제가 선정되어 KTX 경부고속열차 차상컴퓨터의 국산화 개발이 진행되고 있으며 본 과제를 통하여 통신 네트워크 프로토콜의 분석과 열차제어 및 고장현시 기능, 유지보수기능분석, 차상컴퓨터(이후 OBCS[1]) 1편성분 개발과 제작, 성능검증 및 현차시험을 통한 개발품의 안정화를 목표로 활발한 연구가 진행되고 있다.

KTX OBCS는 차량의 각종기능 및 동작을 감시하고 제어 및 통신을 담당하여 기장 및 승무원, 유지보수요원에게 차량의 정보와 상태, 기술지원을 담당하는 중요한 장치이다. 그중 TECA는 동력차의 운전실에 장치되어 열차의 상태와 출발 전 테스트 그리고 고장조치안내 및 훈련지원을 하는 장치로 컴퓨터화된 운전 지원 장치를 말한다.

본 연구과제를 통하여 KTX 운전자 지원단말기인 TECA의 구성과 기능을 알아보고 KTX의 주컴퓨터인 MPU(Main processor unit)와의 통신에 필요한 프로토콜의 분석방법을 살펴봄으로서 KTX OBCS 장치간의 통신프로토콜 분석에 대한 방안을 제시하고자 한다.

---

* 한국철도공사 철도연구개발센터 기술연구팀
** 한국철도공사 수도권철도차량관리단 고속전기팀
*** 한국철도공사 부산고속철도차량관리단 전기팀

1) OBCS : On Board Computer System (고속차량의 제어 및 고장진단을 위한 열차 컴퓨터 장치)

2. 본론

2.1 KTX TECA의 구성

"KTX 운전자 지원단말기 TECA" (이후 "TECA")는 각 동력차 운전실에 1개씩 2개의 모듈 (Display & keypad)로 구성되어 있으며 TECA의 화면제어 및 화면작성을 위한 소프트웨어와 이를 시리얼통신으로 관리하고 up-loading 하는 소프트웨어로 구성되어 있고 Xon/Xoff 통신 프로토콜로 KTX의 주 컴퓨터인 MPU와 상호 통신을 하도록 되어있다.

2.1 하드웨어

TECA의 하드웨어는 그림1과 같이 Display unit과 입력을 담당하는 Keypad가 하나의 구성으로 직렬연결 되어있으며 이는 다시 MPU의 주변장치로 20mA Current loop를 이용한 통신으로 차량의 정보를 운전자에게 현시하게 하는 구조를 이루고 있다.

Display unit은 10.4inch Display 장치와 20mA Current loop card, 386계열의 Microprocessor card 그리고 전원공급장치로 구성되어 있으며 com1 port에 MPU가 com2 port에 Keypad 장치가 연결되도록 구성되어 있으며 Keypad unit은 16개의 Function key 와 10개의 Number key 4개의 Command key로 구성되어 display unit의 J5 커넥터 단자에 RS-232통신을 하는 입력장치이다. 또한 Display unit의 후면에는 5개의 커넥터 단자가 있는데 이들 중 프로그램 설치 및 보안관련 “dongle[2]"장착 25pin port와 data load용 9pin 도 구성되어 있다.

2.2 소프트웨어

TECA의 하드웨어를 구동하고 MPU와의 통신을 위해선 TECA 내에 화면출력용 data와 이를 관리하는 소프트웨어가 있는데 그 종류와 기능은 다음과 같이 구분되어진다.

- TK_LOAD : G0, G4스크린 data up-loading S/W
- TK_TEST : TECA 화면현시상태 Test
- TK_EDIT : 화면작성 S/W
- TK_ESID : TECA의 기본 O/S 설치 S/W
- DAT : DASG1G2.cfi up-loading, Technical Sheet (Zero page, M page) 작성 및 등록 S/W

TECA의 data 관리 및 등록을 위한 소프트웨어를 위해서는 유지보수노트북을 이용하여 운전실 글러브박스 또는 MPU의 KADP 카드 상단에 위치한 9pin serial port에 연결하여 등록하게 되어 있다.

2.3 TECA의 화면구성

G1, G2 스크린은 F1(고장조치안내) 기능관련 화면으로 구성되어 있으며 G0 스크린은 F1 기능 이외의 화면정보를 그리고 G4 스크린은 Technical Sheet(Zero page, M page)로 작성된 스크린 정보를 가지고 있다. 각각의 화면정보는 TECA의 Display unit에 화면 알고리즘은 MPU의 MEMP 카드 내에 존재하여 Display 된다. 각각의 화면정보에는 숫자 및 문자 그리고 Text의 특수기능설정 등의 data가 함께 현시되는 것으로 이에 대한 알고리즘 및 data변경은 Alstom에 의해 수차례 Version-up되어 왔으며 2006년 7월을 마지막으로 data가 등록되었다.

표 1 화면구성 증가

Version-up / 화면의 수	04.3.31	05.4.8	06.7
G1	361	366(↑ 5)	447(↑81)
G2	449	468(↑19)	481(↑13)

이처럼 화면수의 증가와 함께 TECA의 유지보수는 역시 꾸준히 필요한 것으로 나타나고 있으며 이에 대한 분석과 유지보수 기술의 확보는 중요한 문제가 되었다.

---

2) 소프트웨어의 불법복제 및 수정을 막기 위한 Keylock 장치

2.4 TECA의 화면제어

TECA와 MPU는 20mA current loop를 통해 상호간의 정보를 교환하고 화면현시를 위한 정보를 MPU에 의해서 TECA의 Display를 제어하게 된다. 또한 Display unit은 Keypad와 RS-232c Serial line을 통해 29개의 Key 정보를 입력받게 되어 있으며 이는 다시 MPU에 정보를 뿌려주어 필요한 정보를 다시금 Display 하도록 명령을 주는 방식으로 제어된다.

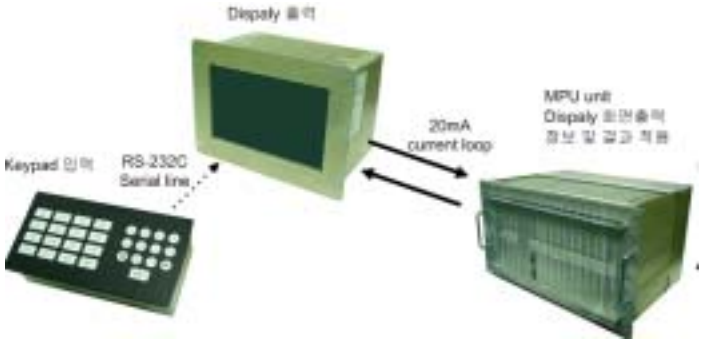

그림 1 TECA의 화면제어방식

3. 통신 프로토콜 분석

3.1 Xon/Xoff 프로토콜

KTX 차량에는 여러 가지 통신프로토콜을 이용하여 OBCS 장치와 하부 주변장치간에 통신을 통하여 기관사와 승무원에게 각종 OBCS 정보가 현시되며, 승객에게는 열차이용에 편의를 제공하게 된다. 그중 기관사에게 정보를 현시하는 장치인 TECA와 주컴퓨터인 MPU의 통신 분석은 중요한 의미를 가지게 된다. MPU의 제어신호와 TECA Display unit과는 20mA current loop를 통해 상호간에 신호를 주고받는 형식을 가지게 되는데 이때 통신을 위한 프로토콜이 Xon/Xoff 프로토콜이다.

표 2 KTX차량에 사용 중인 프로토콜

	MPU	APU	TPU
ATESS	H69010		
RADIO	SHLP		
여객정보현시장치			HBUS
고장현시장치			HBUS
유지보수터미널	XON/XOFF	XON/XOFF	XON/XOFF
기관사콘솔	XON/XOFF		
TMC			XON/XOFF
특수터미널	XON/XOFF		XON/XOFF

Xon/Xoff 프로토콜은 제어 processor 속도와 data 출력의 속도차이나 용량의 차이가 발생할 수 있는 기기간의 통신을 제어하는 프로토콜로 많이 사용되는데 그 예로 프린터의 인쇄명령처리를 볼 수 있겠다. 프린터가 인쇄명령을 받고 이에 대한 data를 처리하는데 있어 프린터의 buffer 용량은 제어 PC의 processor 속도와 처리용량에 비해 작기 때문에 명령이나 data가 가득 차게 되면 제어 PC에 전송 중지명령을 보내게 되고 프린터의 처리가 끝나게 되면 다시금 전송을 요구하여 인쇄를 마치도록 되어 있다.

TECA에 있어서도 MPU와 processor 사이의 속도와 용량의 차이를 극복하고 각 기기간의 통신 data의 정확성과 검증을 위하여 일반적인 패킷의 구조(그림2)와 Checksum을 가지고 있다. 이를 기준으로 TECA의 통신 프로토콜 분석에 기본 구조로 적용하여 보았다.

3.2 프로토콜 분석

통신 프로토콜은 일정한 형식을 가지고 있는데 이를 파악하고 활용하기 위해서는 분석하고자 하는 패킷의 크기와 data section, Checksum 값과 같은 여러 가지 프로토콜 규약을 알아야 한다. 또한 분석을 위한 프로토콜 분석장비와 인터페이스 장비들 그리고 여러 분석의 예를 통해 TECA와 MPU간의 통신 프로토콜 내용을 파악하고 최적화된 분석 방법을 찾아보고 적용해보았다.

그림 2 packet format

앞에서 말했듯이 장치 간에 통신을 파악하고 활용하기 위해서는 이를 분석하는 장비와 기술이 필요하다. 이번 국산화 과제를 진행하는데 있어 차상컴퓨터와 그 주변장치들을 분석하기 위해 구입한 장비가 “LE-7200[3] 프로토콜 분석기”(이하 “분석기”라고 한다)이다. 이 분석기를 통하여 이미 알고 있는 20mA current loop 통신방식에 맞게 OBCS장치와 TECA장치 사이에 인터페이스 장치를 연결하여주고 분석기의 Setup을 설정 한다. 이제 장치간의 통신 data를 Dump하여 이를 분석하는 방법을 찾아 적용해 보도록 하겠다.

3.3 통신 data의 Monitor

TECA는 OBCS의 주컴퓨터인 MPU의 UCVM, MEMP, MELP, KADP 카드들과 자체 저장메모리를 통해 정보를 공유하고 송출하게 된다. 그림2와 같이 시리얼링크 카드인 KADP에 interface card를 장착하여 MPU와 TECA간의 시리얼통신을 연결한 후 분석기의 전원을 인가하면 통신 data의 Monitor를 위한 외부 준비가 끝나게 된다. 이후 차량의 전원을 투입하여 OBCS와 TECA의 정상부팅을 확인한다. 이로서 통신 data의 Monitor 준비를 마치게 된다.

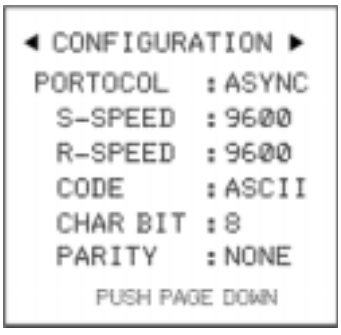

그림 3 analyzer setup

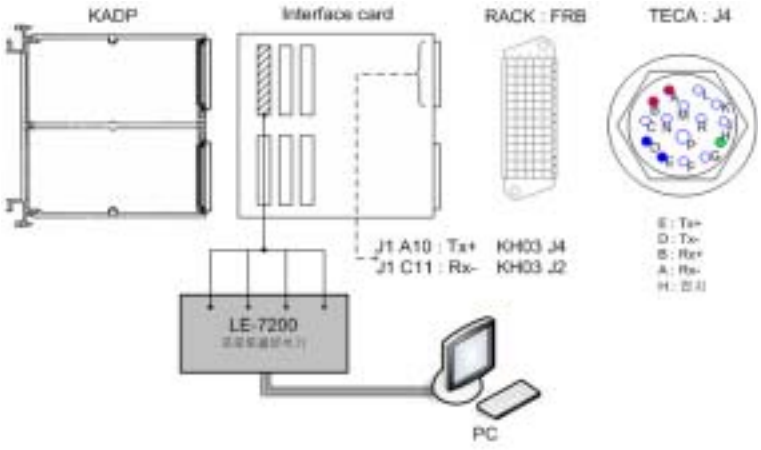

그림 4 통신 data Monitor

분석기의 메모리용량과 분석 data의 크기에 따라 노트북을 연결하고 분석 data를 공유한다면 더 많은 양의 data를 저장할 수 있으며 LE-7200 분석장비와 함께 제공되는 LINEEYE S/W Tool을 이용해 사용자의 PC에서 HEX data를 볼 수 있다. 또한 Text file로 변환이 가능하며 IDLE / TMSP 구분값을 통해 data의 현재시간과 data의 Event 발생에 따른 시간을 1/100초 단위로 현시하는 기능을 통해 data의 비교와 검증에 도움이 된다.

3) LE-7200은 data 통신 중 RS-232C, RS-530(RS-422/485), CURRENT LOOP, TTL, I2C, V.35, RS-449, IrDA, CAN 쪽 data를 캡쳐 하여 분석하기 용이하게 DIGITAL 값으로 표시하여 data 흐름의 이상 유무를 체크 할 수 있으며 내장된 시뮬레이션 기능으로 유지 보수용으로도 사용이 가능한 포토블한 분석장비이다.

3.4 Packet의 분석

TECA와 MPU 사이의 통신프로토콜은 공개되어 있지 않기 때문에 이에 대한 접근은 쉽지 않다. 일반적으로 상용하는 Packet의 format을 통해 Dump된 data를 살펴보고 경우의 수에 대한 TECA의 화면 및 입력 변수를 숫자 정보로 제한하여 통신프로토콜 분석에 접근해 보았다.

분석을 위한 대상으로 TECA의 메뉴 화면구성 중 시간정보를 설정하고 읽어들이는 F4(유지보수기능) 화면으로 하였으며 여기에 시간정보를 설정하는 시간설정(353K087C)화면에서 숫자(시간정보)를 입력함으로써 TECA의 숫자data 전송에 따른 통신 프로토콜을 분석하도록 하겠다. 분석을 위해 첫째 Keypad의 29개 입력값을 분석하여 현시되는 화면들에 따른 프로토콜의 규칙성을 찾았으며, 둘째로 프로토콜 구조 가운데 있는 data section 부분을 분석하였다.

3.4.1 Keypad 입력값 분석

먼저 TECA의 Keypad는 16개의 Function key와 10개의 Number key 그리고 4개의 Command key 를 가지고 있다. 즉 Rs-232C Serial 통신방식으로 TECA Display unit에 29개의 Keypad 입력 값을 전송하게 된다. 이 정보는 다시 MPU에 입력이 되어지며 알고리즘 과정을 거쳐 Display unit에게 화면정보가 출력되도록 진행된다. 이러한 정보의 흐름을 통해 특정 key 값을 입력하고 출력되는 화면을 Monitor 함으로 Key에 대응하는 원하는 프로토콜 data를 수집할 수 있으며 다양한 입력상황을 통해 Key 입력에 대한 규칙적인 패킷의 구조를 파악 할 수 있게 된다.

그림 5는 Keypad 입력 값에 대한 분석기의 HEX code를 분석이 용이한 Text로 변환하여 입력 Key 값에 대한 패킷의 구조를 알아보는 방법을 소개한 것이다.

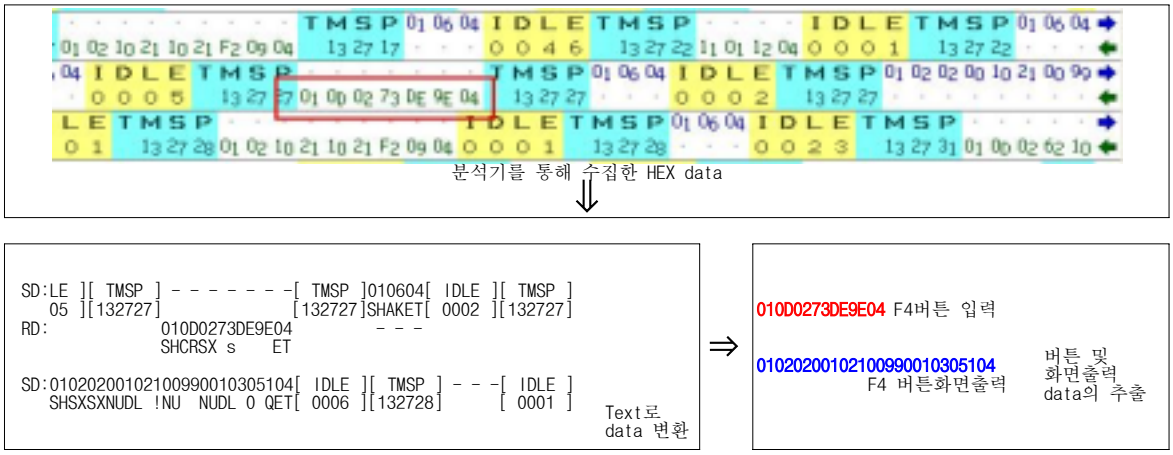

그림 5 data 추출 과정

그림5와 같은 방법으로 29개의 Key입력을 분석한 결과 패킷의 구조는 “010D02 ~ 04”의 구조를 가짐을 알게 되었다. 뿐만 아니라 Key 입력에 대한 Display의 현시 화면에 대한 정보도 함께 얻을 수 있었는데 MPU 화면출력에 대한 패킷의 구조는 “01020200102~04”으로 규칙성을 가짐을 알 수 있다. 이는 화면의 data를 가지고 있는 TECA display unit에 MPU가 화면정보를 현시하도록 명령하는 구조를 가지고 있음도 규명할 수 있었다.

표 3 Keypad의 입력에 따른 Key 값

F1	고장조치안내	010D0270E49B04	F6	열차정보기록	010D0275DAA004	F11	출고전시험	010D027AD0A504
F2	기술자료	010D0271E29C04	F7	제동시험	010D0276D8A104	F12		010D027BCEA604
F3	출고점검	010D0272E09D04	F8	훈련기능	010D0277D6A204	F13	빠져나감	010D0251237C04
F4	유지보수기능	010D0273DE9E04	F9	예비속도계	010D0278D4A304	F14		010D0242416D04
F5	경보발생정보	010D0274DC9F04	F10	언어선택	010D0279D2A404	F15		010D02433F6E04
						F16	메뉴	010D02443D6F04

다음은 TECA의 메뉴화면 버튼을 누르고 시간입력이 진행되는 과정이다. 그림 6은 그 과정과 프로토콜의 이동과정을 나타낸 것이다.

그림 6 시간설정 프로토콜의 이동

3.4.2 data section에 대한 분석

앞에서 29개의 Key값에 대한 패킷의 구조를 찾아볼 수 있었다. 그러나 시간정보입력에 따른 변수 data는 data section에 더 많은 정보를 가지게 되며 이에 따른 복잡한 구조를 가지게 된다. F4 유지보수기능의 시간설정 화면은 숫자변수에 대한 data section의 구조를 분석할 수 있는 예를 제공한다.

분석과정은 다음과 같다. TECA의 시간설정화면에 특정 sample 시간변수를 입력하면 이 정보는 MPU의 KADP card를 거쳐 UCVM card에 저장된다. UCVM에 기록된 정보는 다시금 KADP를 통해 TECA로 화면출력 신호로 보내지게 되며 이 과정을 Monitor 함으로 숫자 data에 대한 기본 정보와 시간정보, 그리고 자리 변수를 가진 패킷의 기본구조를 알 수 있게 된다. 그림 7은 data section이 포함된 프로토콜의 크기와 이동을 볼 수 있다.

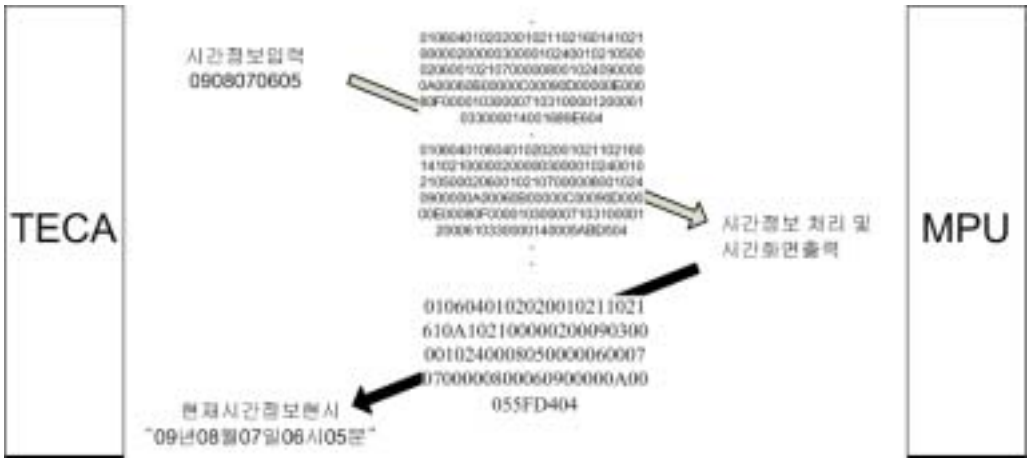

그림 7 data section 정보이동

이에 따라 F4 시간입력화면에서 시간정보 “09년 08월 07일 06시 05분”을 입력하여 보았으며 이에 대한 keypad의 입력값을 적용함으로 Text화된 프로토콜 data를 분석하였다. 이와 같은 방법으로 시간정보의 값을 임의 설정하여 입력하여 data section의 시간입력 변수와 자리값에 대한 정보를 추출할 수 있었다. Sample 값으로는 시간입력의 오류를 최소화하고 반복되는 수와 규칙성을 두었으면 0~9까지의 입력변수가 포함된 시간입력변수를 사용하였다.

“0908070605” 시간입력에 대한 값을 받게 되면 다음과 같다. “0106040102020010211021610A1021000002000903000010240008050000060007070000080006090000 0A00055FD404” 여기에 앞에서 설명한 packet format을 적용하여 Key 값과 화면정보를 통해 알게 된 start flag “010604”과 Address & Control에 해당되는 “01020200102110216 10A1021”은 모든 Sample 값에서 동일하게 적용되었으며 End flag 값에 “04”을 제외하면 data section과 Checksum 값만 남게 된다. Data section은 Sample 값에 대한 data를 조합하여 규칙성과 반복되는 부분들을 적용하여보았다.

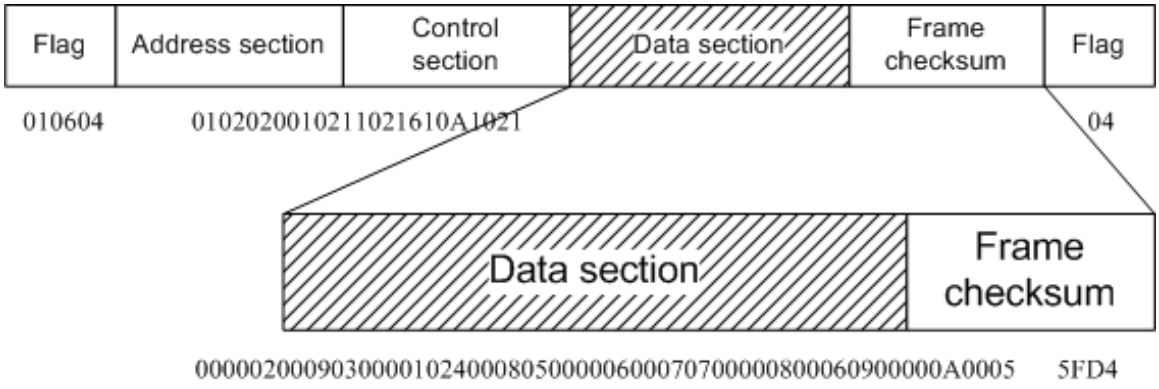

그림 8 Data section & Checksum

이와 같이 data section 과 Checksum 값만을 정리할 수 있게 된다. 여기에 입력된 프로토콜 data 가운데 반복 및 규칙성을 가지는 정보들을 구분하여 정리하여 보았을 때 시간입력 값과 시간입력을 위한 data field 값도 알게 된다.

표 4 시간입력 값과 data field 값

1111111111 시간입력	0403020100시간입력	0908070605시간입력
00 **1021** 02	00 **00** 02	00 **00** 02
00 **1021** 03	00 **1024**03	00 **09** 03
00 **1021** 1024	00 **00** 1024	00 **00** 1024
00 **1021** 05	00 **03** 05	00 **08** 05
00 **1021** 06	00 **00** 06	00 **00** 06
00 **1021** 07	00 **02** 07	00 **07** 07
00 **1021** 08	00 **00** 08	00 **00** 08
00 **1021** 09	00 **1021**09	00 **06** 09
00 **1021** 0A	00 **00** 0A	00 **00** 0A
00 **1021** EB61	00 **00** BE8E	00 **05** 5FD4

앞에서와 같이 정리하여 시간입력 변수에 대한 프로토콜 값들을 찾을 수 있게 되었으며 이에 대한 시간입력 data filed 값도 찾을 수 있었다. 이를 다시한번 정리하면 다음과 같다.

표 5 data section & Checksum 값 정리

10진수	시간변수
0	00
1	1021
2	02
3	03
4	1024
5	05
6	06
7	07
8	08
9	09

자리설정 값	
1	02
2	03
3	1024
4	05
5	06
6	07
7	08
8	09
9	0A
10	Checksum

시간설정 값	Checksum
1111111111	EB61
0403020100	EB8E
0908070605	5FDA
2210222234	86BC
3416252259	F63F
6010161617	0E2C
7810091718	FC30

4. 프로토콜의 활용과 개발 방향

얻어진 data의 변수들을 분석장비의 Simulation 기능으로 Display unit에 재전송하여 분석을 통해 얻어진 변수들에 대한 검증을 할 수 있으며 검증된 화면정보를 이용하여 기관사 지원화면의 수정 및 추가할 수 있는 기술에 접근 할 수 있게 된다. 이는 변화하고 다양화되는 고장유형에 따른 GDI 수정과 추가기능에도 적용 될 것이라고 본다. 물론 TECA의 프로토콜분석이 모두 마쳐진 것이 아니다. 앞으로 분석해야할 Text data의 구조와 영문과 한글에 따른 기능의 변화, 기능키와 특수 화면에 대한 제작과 통신 프로토콜의 알고리즘 분석, 유지보수용 S/W 개발과제도 남아있다. 뿐만 아니라 동일 Xon/Xoff 프로토콜을 사용하는 OBCS 주변기기에 대한 패킷의 분석과 기타 HBUS, H69010, SHLP 프로토콜을 사용하는 장치의 분석까지 많은 연구과제가 남아있다.

5. 결론

TECA 개발은 하드웨어의 성능향상과 함께 고속열차의 운전규정 변경과 문구수정 및 고장조치안내(GA)와 고장조치안내서(GAA)의 400페이지 항목 통일 등의 목적으로 수차례 제작사인 Alstom측에 요구하여 기능개선을 해왔다. 그러나 향후 지속적인 유지보수와 기능 개선을 위해서는 TECA의 기술개발이 필요하며 이에 따른 TECA의 설계, 제작기술과 소프트웨어 및 통신 인터페이스의 알고리즘분석은 자체적인 운영기술을 한단계 업그레이드 하는데 크게 이바지 할 수 있으며 뛰어난 IT 기술과 Display 기술을 가진 우리나라의 인프라를 통해 보다 향상된 선진기술을 창출할 수 있으리라 본다.

참고문헌

[1] 차상컴퓨터 1, 철도경영연수원, 2002
[2] MULTI PROTOCOL ANALYZER 매뉴얼, LE정보통신, 2004.
[3] 기관사화면지원, KTX차량컴퓨터제어장치개발, 2006