**New CIT TPCL Specification**

**( New CIT Interface Protocol )**

**TITENG**

|  |  |  |  |
| --- | --- | --- | --- |
| **개정 이력** | | | |
| **날짜** | **개정** | **설명** | **작성자** |
| 2009-12-07 | Rev 0.1 | 초안 작성 – 기존 CIT TPCL Version 1.1.19 기준 적용  추가.변경 사항은 “**RED**” color로 표시함. | 황재호 |
| 2009-12-14 | Rev 0.2 | 명령어 체계 변경 – Command/Response Frame format 변경.  RED color 제거, Diagnostic, Test mode 추가 검토  Parameter로 사용되는 값들 중 MS관련 data를 제외하고는 ASCII code값을 숫자로 변경함 | 황재호 |
| 2009-12-24 | Rev 0.3 | 명령어 체계 변경 – 명령어 통합 (Color, Mono, Varnish print 명령어 통합)  그 외 통합 가능한 명령어들을 통합 / 명칭 변경.  추가 – Ribbon Sync, Card Size설정, ….. | 황재호 |
| 2009-12-30 | Rev 0.4 | 명령어 추가 및 그에 따른 검토– “!GYP | 황재호 |
| 2010-05-11 | Rev 0.5 | 명령어 추가/삭제 – 매개변수 중 일부 OLD TPCL과 호환 유지 | 구순자 |
| 2010-05-27 | Rev 0.6 | RF 명령어 세분화 및 매개변수 변경 | 구순자 |
| 2010-06-08 | Rev 0.7 | Mono Print 데이터 구조 구체화 | 황재호 |
| 2010-06-16 | Rev 0.8 | RF-TAG 명령어 추가 | 황재호 |
| 2010-06-22 | Rev 0.9 | IC – New Command(ICS, IPO, ISA, IPF) 내용 일부 변경  RF – New Command(RMV, RCC, RPO, RSA, RPF, RMK, RMR, RMW, RMI, RMD) 내용 일부 변경 | 구순자 |
| 2010-06-24 | Rev1.00 | $STT 명령어 추가 | 황재호 |
| 2010-08-26 | Rev 1.01 | USB Description과 Serial Number 읽기/쓰기 명령어 추가  (GDS, SDS, GSN, SSN) | 구순자 |
| 2010-09-20 | Rev 1.02 | MS JITTER 명령어 추가 (&MRJ, &MWJ)  LCD message save 명령어 추가 (+LMS) | 김혜진 |
| 2010-11-09 | Rev 1.03 | 2차 B/D의 DIP Switch 삭제에 따른 Parameter변경.  System Information 읽기와 설정 parameter재 설정. - !GOI, !AOI  Laminator interface 명령어 추가 | 황재호 |
| 2010-11-16 | Rev 1.04 | Laminator interface를 위한 인쇄 명령의 Parameter변경 | 황재호 |
| 2010-11-23 | Rev 1.05 | Laminator interface 명령의 Parameter변경 | 황재호 |
| 2010-11-24 | Rev 1.06 | 인쇄 시 Laminator 관련 option 변경 - $PCP, $SPT  $POL 명령 추가 | 황재호 |
| 2010-11-29 | Rev 1.07 | &LCR의 option추가 | 황재호 |
| 2010-12-14 | Rev 1.08 | USB Manufacturer 읽기/쓰기 명령어 추가 (GMF, SMF) | 구순자 |
| 2010-05-23 | Rev 1.09 | 인쇄명령($CCB, $TCB, $PCP) 변경 – 전달되는 이미지 Format 추가  $CDT, !RTL, !RTU 명령 추가 | 황재호 |
| 2010-06-02 | Rev1.10 | “$IFO” 명령 삭제, | 황재호 |
| 2010-06-02 | Rev1.11 | “$EOP”, “$CPJ”, “$PJS”, “$RPE” 명령 추가, Error45추가 | 황재호 |
| 2011-08-19 | Rev1.13 | !CFS 명령어 응답 데이터 변경 | 윤성중 |
| 2011-08-24 | Rev1.14 | !RSP, !RCL, !RCR 명령 추가 | 윤성중 |
| 2011-09-01 | Rev1.15 | $PJS 명령어(print job status) 응답 parameter추가  $PCP 명령의 parameter 추가 적용. | 황재호 |
| 2012-02-27 | Rev1.16 | 자동 장애 처리 옵션 추가 (!GCV, !SCV 항목 38 추가)  자동 장애 처리시 프린터 상태 확인 ($PJS 명령어 응답 parameter추가)  On Time 테이블 포맷 수정 (Color: 16->32bit, Mono: 4->8bit)  Panel (Color/Mono) data transmit format 수정 (Color: 16->32bit, Mono: 4->8bit)  -“$STT”, “$TCB” parameter 재설정 | 박영진 |
| 2012.04.18 | Rev1.17 | 디지털 가변 저항 레벨 값(0~255) 세팅 하기 (!SDP)  저장된 디지털 가변 저항 레벨 값 (0~255) 불러오기 (!GDP) | 박영진 |
| 2012.12.27 | Rev1.18 | !REH 명령어 Parameter 추가  !GCV와 !SCV의 Parameter 추가(39 ~ 41)  !SDV 명령어 추가 – DC Motor Duty Value 설정  !GDV 명령어 추가 – DC Motor Duty Value 읽기 | 윤성중 |
| 20130108 | Rev1.19 | !SPS 명령어 추가 – 장비 시리얼 번호 설정  !GPS 명령어 추가 – 장비 시리얼 번호 읽기 | 지정환 |
| 20131016 | Rev.1.20 | !CRC 명령어 추가 - 남은 리본의 개수 | 조성제 |
| 20130123 | Rev.1.21 | !GCV와 !SCV의 Parameter 41(ribbon maker) value 수정 (0:DPS, 1:DNP, 2:ITW) | 지정환 |
| 20130225 | Rev.1.22 | !SLF와 !DFW 의 명령어 추가 | 조성제 |
| 20130307 | Rev.1.23 | !SRS 명령어 추가 | 지정환 |
| 20130308 | Rev.1.24 | !SCB 명령어 추가 - LCD Contrast, RF, RF-TAG, Changer, IFM Board  Power Control  !SCF 명령어 추가 – Backup Configure Data 설정  !LCF 명령어 추가 – Load Configure Data 설정 | 조성제 |
| 20130419 | Rev.1.25 | !DFE 명령어 추가 - Download F/W Erase  !DFD 명령어 추가 - Download F/W Data Write  !DFV 명령어 추가 - Download F/W Verify  !DFR 명령어 추가 - Download F/W Reset  !DFS 명령어 추가 - Download F/W Get Status | 조성제 |
| 20130702 | Rev.1.26 | !SIA 명령어 추가 – Set IP Address  !GIA 명령어 추가 – Get IP Address | 지정환 |
| 20130703 | Rev.1.27 | $PCP 명령어 변경  $CDT 명령어 변경 | 정진수 |
| 20130724 | Rev.1.28 | !GRS 명령어 변경 (Response Parameter 추가)  !SSC 명령어 추가  !GSC 명령어 추가  !SRR 명령어 추가 - Set RF-Tag Ribbon Type  !GRR 명령어 추가 - Get RF-Tag Ribbon Type | 조성제 |
| 2013-08-28 | Rev.1.29 | !DFF 명령어 추가 | 윤성중 |
| 2013-09-13 | Rev.1.30 | $PCP 명령어 변경(인쇄 속도/ Data 전송 모드 Parameter 추가) |  |
| 2013-09-13 | Rev.1.31 | !LRM 명령어 추가  !SRM 명령어 추가 | 김원영 |
| 2013-10-22 | Rev.1.32 | !GCV , !SCV, $SPT, !CRS, !SRT – Gold/White/Silver Ribbon 파라미터 추가. | 김원영 |
| 2013-12-17 | Rev.1.33 | !STD,!EBC 명령어 추가. $PCP 파라미터 추가. | 김원영 |
| 2014-05-20 | Rev.1.34 | !GCV,!SCV – 카드입력 옵션 Parameter 추가. | 김원영 |
| 2014-05-28 | Rev.1.35 | !GCV , !SCV, $SPT, !CRS, !SRT – Red/Blue Ribbon 파라미터 추가. | 조성제 |
| 2014-06-27 | Rev.1.36 | !SCV, !GCV, !LRM, !SRM | 조성제 |
| 2014-6-30 | Rev.1.37 | !SEI | 조성제 |
| 2014-7-09 | Rev.1.38 | !GCV,!SCV – 디스펜서 옵션 Parameter 추가. | 김원영 |
| 2014-08-05 | Rev.1.39 | !EBC,$PCP 옵션 Parameter 수정. | 김원영 |
| 2014-8-11 | Rev.1.40 | !PSS 명령어 추가(Only Kiosk 향) | 윤성중 |

**Contents**

[!STD,!EBC 명령어 추가. $PCP 파라미터 추가. 4](#_Toc395616817)

[!GCV,!SCV – 카드입력 옵션 Parameter 추가. 4](#_Toc395616818)

[**1. New CIT TPCL Programmer’s Guide** 10](#_Toc395616819)

[**1.1 Image memory arrangements** 10](#_Toc395616820)

[**1.2 Panel (color/mono) data transmit format** 11](#_Toc395616821)

[1.2.1. **default format** (1차 적용 format ) 11](#_Toc395616822)

[1.2.2. **Original RAW format** (2차 적용 format) 13](#_Toc395616823)

[1.2.3. **Speed\_Image\_Optimized RAW format** (3차 적용 format) 14](#_Toc395616824)

[**1.3 Image compression method** 15](#_Toc395616825)

[**2. Command/Response Syntax** 16](#_Toc395616826)

[**Syntax:** all command & response data transfer 16](#_Toc395616827)

[**2.1 Basic command syntax** 16](#_Toc395616828)

[**2.2 Basic response syntax** 16](#_Toc395616829)

[3. **Command List** 17](#_Toc395616830)

[4. **Command Description** 23](#_Toc395616831)

[**4.1 General Command** 23](#_Toc395616832)

[!RRP Command – Run Reset Printer 23](#_Toc395616833)

[!RCE Command – Run Clear Error status 23](#_Toc395616834)

[!RBZ Command – Run Buzzer Control 23](#_Toc395616835)

[!RIP Command – Run Input card to Print ready position 24](#_Toc395616836)

[!REH Command – Run Exit loaded card to output Hopper position 24](#_Toc395616837)

[!RTC Command – Run Turn up a Card 24](#_Toc395616838)

[!RRS Command – Run Ribbon Sync & Skip 24](#_Toc395616839)

[!RCL Command – Run Cleaning The printer 25](#_Toc395616840)

[!RSP Command – Run Self-Printing 25](#_Toc395616841)

[!RCR Command – Run Card Reject 25](#_Toc395616842)

[!CFS Command – Check Feeder Sensors 26](#_Toc395616843)

[!CRS Command – Check Ribbon Sensors 26](#_Toc395616844)

[!CRC Command – Check Ribbon Remained Count 26](#_Toc395616845)

[!CSS Command – Check Status and Sensors 27](#_Toc395616846)

[!GVT Command – Get CIT Version / Type 27](#_Toc395616847)

[!GRS Command – Get Ribbon Sensor Value 27](#_Toc395616848)

[!SRS Command – Get Ribbon Sensor Value 27](#_Toc395616849)

[!GOI Command – Get Option Information Value 28](#_Toc395616850)

[!AOI Command – AS(factory) set Option Information Value 28](#_Toc395616851)

[!GSI Command – Get System Information Value 30](#_Toc395616852)

[!ASI Command – AS(factory) set System Information Value 30](#_Toc395616853)

[!GCV Command – Get CIT(printer) parameter Value (각 항목별) 31](#_Toc395616854)

[!SCV Command – Set CIT(printer) parameter Value (각 항목별) 31](#_Toc395616855)

[[Appendix] CIT(printer) Parameter List (For GCV, SCV command) 32](#_Toc395616856)

[!SRT Command – Set Ribbon Type 34](#_Toc395616857)

[!GPC Command – Get Print Count 34](#_Toc395616858)

[!SPC Command – Set Print Count (생산/개발에만 사용 가능) 34](#_Toc395616859)

[!GMA Command – Get MAC Address 36](#_Toc395616860)

[!SMA Command – Set MAC Address (생산/개발에만 사용 가능) 36](#_Toc395616861)

[!ASR Command – AS/Factory Set RF-TAG Valid No (생산/개발에만 사용 가능) 37](#_Toc395616862)

[!ARR Command – AS/Factory Read RF-TAG Valid No 37](#_Toc395616863)

[!GMF Command – Get USB Manufacturer 38](#_Toc395616864)

[!SMF Command – Set USB Manufacturer (생산/개발에만 사용 가능) 38](#_Toc395616865)

[!GDS Command – Get USB Description 38](#_Toc395616866)

[!SDS Command – Set USB Description (생산/개발에만 사용 가능) 38](#_Toc395616867)

[!GSN Command – Get USB Serial Number 39](#_Toc395616868)

[!SSN Command – Set USB Serial Number (생산/개발에만 사용 가능) 39](#_Toc395616869)

[!RTL Command – Request Terminal Lock 39](#_Toc395616870)

[!RTU Command – Release Terminal Lock 39](#_Toc395616871)

[!SDP Command – Set Digital Potentiometer 39](#_Toc395616872)

[!GDP Command – Get Digital Potentiometer 40](#_Toc395616873)

[!SDV Command – Set Ribbon DC Motor Duty Value 40](#_Toc395616874)

[!GDV Command – Get Ribbon DC Motor Duty Value 41](#_Toc395616875)

[!SPS Command – Set Printer Serial Number 41](#_Toc395616876)

[!GPS Command – Get Printer Serial Number 41](#_Toc395616877)

[!SLF Command – Set Loader Flag 41](#_Toc395616878)

[!DFW Command – Download FirmWare 42](#_Toc395616879)

[!SCB Command – Set Power Control Board 42](#_Toc395616880)

[!SCF Command – Savd Configure Data 42](#_Toc395616881)

[!LCF Command – Load Configure Data 43](#_Toc395616882)

[!DFE Command – Download Firmware Erase 43](#_Toc395616883)

[!DFD Command – Download Firmware Data Write 43](#_Toc395616884)

[!DFV Command – Download Firmware Verify 43](#_Toc395616885)

[!DFR Command – Download Firmware Reset 44](#_Toc395616886)

[!DFS Command – Download Firmware Get Status 44](#_Toc395616887)

[!DFF Command – Download Flipper Firmware 44](#_Toc395616888)

[!STD Command – Send To Dispenser 44](#_Toc395616889)

[!EBC Command – Emission Back Card 45](#_Toc395616890)

[!SIA Command – Set IP Address 45](#_Toc395616891)

[!GIA Command – Get IP Address 45](#_Toc395616892)

[!SSC Command – Set Ribbon Sensor Change Flag 46](#_Toc395616893)

[!GSC Command – Get Ribbon Sensor Change Flag 46](#_Toc395616894)

[!SRR Command – Set RF-TAG Ribbon Type 46](#_Toc395616895)

[!GRR Command – Get RF-TAG Ribbon Type 47](#_Toc395616896)

[!LRM Command – Load Ribbon Maker (New Command) 47](#_Toc395616897)

[!SRM Command – Set Ribbon Maker (New Command) 47](#_Toc395616898)

[!SEI Command – Set EEPROM Init (New Command) 47](#_Toc395616899)

[!PSS Command – Printer Sensor Status(Only Kiosk 향) 48](#_Toc395616900)

[**4.2 Option Command** 49](#_Toc395616901)

[&MEC Command – MS set Encoder Coercivity 49](#_Toc395616902)

[&MST Command – MS Set(write) Track buffer 49](#_Toc395616903)

[&MWT Command – MS Write Track data 49](#_Toc395616904)

[&MRT Command – MS Read Track data 50](#_Toc395616905)

[&GBT Command – Get IFM and SAM B/D type (Check GemCore Serial and Pro) 51](#_Toc395616906)

[&CSI Command – Check SIM sensor 51](#_Toc395616907)

[&CSA Command – Check SAM list 51](#_Toc395616908)

[&IMC Command – IC Move Smart Card to Docking Station 52](#_Toc395616909)

[&ISC Command – IC Send Command to Smart Card 52](#_Toc395616910)

[&SAM Command – SAM Send Command 52](#_Toc395616911)

[&PS1 Command – PSAM Send Command 53](#_Toc395616912)

[&PS2 Command – PSAM2 Send Command 53](#_Toc395616913)

[&RMP Command – RF Card MoveContactless Position 54](#_Toc395616914)

[&RSC Command – RF Send Command to RF card 54](#_Toc395616915)

[[Appendix] IC/SAM Command/Response Format 55](#_Toc395616916)

[[Appendix] PSAM Command/Response Format 56](#_Toc395616917)

[[Appendix] RF Command/Response Format 56](#_Toc395616918)

[&ICS Command – IC Change Slot Number (New Command) 57](#_Toc395616919)

[&IPO Command – IC Power On (New Command) 57](#_Toc395616920)

[&ISA Command – IC Send Command (New Command) 57](#_Toc395616921)

[&IPF Command – IC Power oFf (New Command) 58](#_Toc395616922)

[&MRJ Command – read Jitter data after Reading MS data 58](#_Toc395616923)

[&MWJ Command – read Jitter data after Writing and reading MS data 58](#_Toc395616924)

[&RMV Command – RF Get Module Version (New Command) 59](#_Toc395616925)

[&RCC Command – RF Check Card (New Command) 59](#_Toc395616926)

[&RPO Command – RF Power On (New Command) 59](#_Toc395616927)

[&RSA Command – RF Send Command (New Command) 60](#_Toc395616928)

[&RPF Command – RF Power Off (New Command) 60](#_Toc395616929)

[&RMK Command – RF Set Key (Mifare) (New Command) 61](#_Toc395616930)

[&RMR Command – RF Read Data (Mifare) (New Command) 61](#_Toc395616931)

[&RMW Command – RF Write Data (Mifare) (New Command) 61](#_Toc395616932)

[&RMI Command – RF Increment Block (Mifare) (New Command) 62](#_Toc395616933)

[&RMD Command – RF Decerement Block (Mifare) (New Command) 62](#_Toc395616934)

[**4.3 Laminator Command** 63](#_Toc395616935)

[&LRP Command – Laminator Read Paramters (New Command) 63](#_Toc395616936)

[&LSP Command – Laminator Set Paramters (No write into flah) 63](#_Toc395616937)

[&LWP Command – Laminator Write Paramters (Write into flah) 64](#_Toc395616938)

[&LMW Command – Laminator CIT 🡨🡪 Laminator interface working mode 64](#_Toc395616939)

[&LCR Command – Laminator command: Status Get Request 65](#_Toc395616940)

[&LCI Command – Laminator command: Initialize Request 66](#_Toc395616941)

[&LCW Command – Laminator command: Warming-Up (Card entry wait) 67](#_Toc395616942)

[&LCL Command – Laminator command: Laminating Proceess 67](#_Toc395616943)

[&LCE Command – Laminator command: Laminating End Proceess 67](#_Toc395616944)

[&LCS Command – Laminator command: Laminator Save Mode 68](#_Toc395616945)

[&LCJ Command – Laminator command: Card Eject 68](#_Toc395616946)

[&LCY Command – Laminator command: Laminator Ribbon Sync 69](#_Toc395616947)

[**4.4 Print Command** 70](#_Toc395616948)

[$STT Command – Set On-Time Table 71](#_Toc395616949)

[$SOP Command – Start Of Printing 72](#_Toc395616950)

[$SPT command – Set the print type 73](#_Toc395616951)

[$CCB command – Clear Color/Varnish/Mono Image Buffer 73](#_Toc395616952)

[$TCB command – Color/Varnish/Mono Image Data Transfer (with parameters) 74](#_Toc395616953)

[$PCP command – Print Color/Varnish/Mono Panel (a Panel / All Panel) 75](#_Toc395616954)

[$POL command – Print Only Laminator (No printing) 76](#_Toc395616955)

[$CDT command – Calibrate Data Transfer 76](#_Toc395616956)

[$EOP Command – End Of Print 77](#_Toc395616957)

[$CPJ Command – Cancel Print Job 77](#_Toc395616958)

[$PJS Command – Print Job Status 77](#_Toc395616959)

[$RPE Command – Re-Print start after Error 78](#_Toc395616960)

[4.5 SetUp Command 79](#_Toc395616961)

[+LDD Command – LCD Display Data (16 character/line) 79](#_Toc395616962)

[+LDO Command – LCD Display Option command 79](#_Toc395616963)

[+LUR Command – LCD User memory Read (Read LCD memory) 80](#_Toc395616964)

[+LUW Command – LCD User memory Write (Write data into LCD memory) 80](#_Toc395616965)

[[Appendix] 사용자 LCD 읽기/쓰기에 사용할 Text 또는 Image 파일 구조 81](#_Toc395616966)

[+UMC Command – Clear User’s Memory 82](#_Toc395616967)

[+UMR Command – Get(Read) User’s Memory 82](#_Toc395616968)

[+UMW Command – Write User’s Memory 83](#_Toc395616969)

[5. **Status / Error Code** 84](#_Toc395616970)

[**5.1 Status Code** 84](#_Toc395616971)

[**5.2 Error Code** 84](#_Toc395616972)

[**5.3 DLL Error Code** 89](#_Toc395616973)

# **1. New CIT TPCL Programmer’s Guide**

## **1.1 Image memory arrangements**

**MCj03120920000[1]**



## **1.2** **Panel (color/mono) data transmit format**

인쇄 데이터 구조와 데이터 전송 방법은 3가지로 나눠진다.

이미지의 가로pixel 값이 32의 배수로 설정되지 않을 시는 0(null)을 채워 32bytes의 배수로 만든다.

세로는 768pixel line으로 fix되어 전송된다.

### 1.2.1. **default format** (1차 적용 format )

**- Color 32bits no-rotate / Mono 8bits no-rotate format**



Driver / DLL에서는 “1024 X 768”의 이미지를 만들고, 이미지의 pixel resolution은 256 level의 data를 사용한다.

New-CIT(TP9000/9100)로 Data 전송 시 Data format은 32bit format으로 변경한다.

1 pixel의 data는 4bytes로 만들어져, Line단위로 전송된다.

전송되는 Line의 순서는: 1 – 385 – 2 – 386 – 3 – 387 – 4 – 388 – 5 – 389 - ……..…….. – 382 – 766 – 383 – 767 – 384 – 768

< 32bit pixel data 적용 이유>

* CIT 이미지 품질 개선을 위한 Control bit를 사용한다. (256,1024계조)
* 이미지 개선을 위해, TPH On-time조절을 세밀하게 조절하기 위해 최소 12bit를 사용한다.
* TPH의 잔열 발생을 최소화 하기 위해, 최소 마지막 2bit는 ‘0’으로 처리, TPH를 Off한다.

\* 전송되는 **Color data의 format**은 아래 그림과 같은 구조로 된다.



만들어지는 Color Data는 가로( 위 그림의 1024pixels )는 32의 배수로 설정되어야 한다. (32bytes , 64bytes, ….)

세로는 768 pixel line으로 fix되어 전송 된다.

\* 전송되는 **Mono data의 format**은 아래 그림과 같은 구조로 된다.

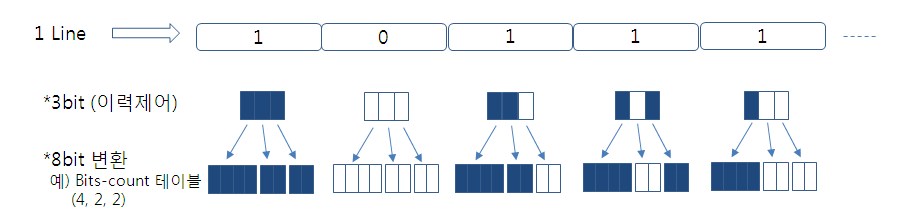


Mono panel (Varnish)의 Data format 은 8bit를 기본으로, Color와 같은 구조로 Data가 전송된다.

Data size는 전체 full로 적용하였을 때, 768KB를 사용한다.

**Mono Data는 기본 1pixel당 8bits로, 가로는 MAX 1024bytes로 32의 배수로 설정되어야 한다 (128, 256, 320, 512bytes)**

1 Line data format은 다음 그림과 같다.



“**$STT**”명령을 통해 8bit 변환 테이블을 정의하여 3bit를 8bit로 변환한다

(단 8bit Bits-count 테이블의 값들의 합은 8이 되도록 구성한다.).

“!GCV” 명령을 통해, SET(Printer)의 Y-axis value를 사용하여 Image의 선단의 Null Line count가 결정된다.

### 1.2.2. **Original RAW format** (2차 적용 format)

**- Color 8bits no-rotate / Mono 1bits no-rotate format**

Driver / DLL에서는 “1024 x7 68”의 이미지를 만들고, 이미지의 pixel resolution은 256 level과 1level의 data 사용.

Data 전송 시 original data를 전송한다.



Color와 마찬가지로, Mono에서도 순서대로, 1bit data를 전송한다.

Mono = 96K bytes (128 bytes(1024/8) \* 768 line)

FW에서 rotate와, 이미지 처리를 모두 수행할 경우 사용하는 format으로, 시간이 늦어진다.

### 1.2.3. **Speed\_Image\_Optimized RAW format** (3차 적용 format)

**- Color 32bits Rotate / Mono 8bits Rotate format**

Default format과 비슷하나, PIXEL data가 rotate된 format이다.

PC로부터 전송되는 Data는 인쇄 순서대로 Data가 전송되고, 장치(CIT)에서는 별도의 rotate와 TPH 이미지 전송 format을 맞추기 위한 작업을 수행하지 않는다.

Data(dot 및 line 별) history 제어는 PC-Driver수행하고, PC에서 만들어진 데이터를 전송한다.

장치(CIT-FW)에서는 각 컬러별/라인별 에너지 제어를 위한 정보를 별도의 명령을 통해 PC로부터 전송 받는다.

아래 그림과 같이 **1st 96 bytes – 2nd 96bytes - ….. 16th 96 bytes - …**. 의 순으로 data가 전달되고, 메모리에 쌓인다.



각 Pixel 라인의 에너지 제어를 위한 에너지 명령은 “$CDT” 명령을 사용한다.

“**$CDT**” 명령은 데이터 전송 format이 **”Speed\_Image\_Optimized RAW format”**일 경우만 사용한다.

Mono 역시 rotate된 format으로, 전송된다.

## **1.3 Image compression method**

- Compress method는 기본적으로 사용하지 않는다.

- 기존 CIT에서 사용한 run length method는 사용하지 않는다.

- 사용 가능한 compress method는 image data를 그대로 전달 될 때 사용한다.

- 인쇄 명령 중 “$TCB”에서 적용되는 Data Mode는

**RAW**: 압축되지 않은 인쇄 data

**JBIG**: JBIG image data – DC1250에서 decoding지원, 적용 방법은 추후 고려한다 (각 PANEL별)

**JPEG**: JPEG image data – DC1250에서 decoding지원, 적용 방법은 추후 고려한다 (Image)

# **2. Command/Response Syntax**

## **Syntax:** all command & response data transfer

* ESC : 0x1B, CR: 0x0D
* Command/Response Frame : 0x1B (Cmd/Rsp) (Status) (length) (Data) 0x0D

## **2.1 Basic command syntax**

아래 그림의 “data”는 Command에 따른 Parameter와 추가 필요한 data를 가진다.

**Length는 4bytes로 “DATA” field의 Length이다**.



명령어는 4가지 그룹으로 나뉘어져 있다.

Command의 4bytes중 앞 첫 byte는 명령어 종류를 나타낸다.

**!**XXX – General Command (Diagnostic, General mecha. Control, Parameter설정/확인)

**&**XXX – Option Command (MS / IC / RF / SAM 등 Option으로 추가되는 기능에 대한 Command)

**$**XXX – Print Command (인쇄 관련한 명령어)

**+**XXX – Setup Command (장치 설정 관련한 명령어 – LCD, Memory 설정/확인)

Ex) $DCG 명령어 6570 bytes data (parameter + real data) 사용 시

: 1B $DCG 000019AA <Data 6570(0x19AA) bytes> 0D

!MIC 명령어 0 bytes data

: 1B !MIC 00000000 0D

## **2.2 Basic response syntax**

아래 그림의 Response는 항상 “Status”를 응답한다.

“data”는 Command에 따른 Parameter와 추가 필요한 data를 가진다.

**Length는 4bytes로 “DATA” field의 Length, status는 CIT의 내부 status이다**.



명령에 대한 응답은 Response field로 적용 되고, status는 응답에 대한 상태 및 현재 상태를 나타낸다.

응답 중 DATA가 없어도, Length field는 ‘0’으로 나타낸다.

응답 종류 역시 4가지 그룹으로 나뉘어지고, 명령의 종류에 따른 응답을 구분한다.

**!** 🡪 **@** (General), **&** 🡪 **\*** (Option), **$** 🡪 **%** (Print), **+** 🡪 **=** (Setup)

Status는 Error (Error Code – timeout, jam, …) / Information (Running) / OK, 등을 나타낸다.

Ex) !CRS의 10 bytes 응답(22bytes): 1B @CRS 0000 0000000A <Data 10 bytes> 0D

# 3. **Command List**

**1) General Command (!XXX - Diagnostic, General mecha. Control, Parameter설정/확인)**

R: Run(Mecha.구동), C: Clear/Check, G: Get Info, S:Setting, D:Diag Command, A: AS or Factory set

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Data** | **Description** | **CIT** |
| !RRP |  | **R**un **R**eset **P**rinter | R |
| !RCE |  | **R**un **C**lear **E**rror status | . |
| !RBZ | p1 p2 p3 | **R**un **B**u**z**zer control | BZ |
| !RIP |  | **R**un **I**nput card to **P**rint position | MI |
| !REH | p1 | **R**un **E**xit loaded card to output **H**opper position | MO,MOX |
| !RTC |  | **R**un **T**urn up a **C**ard (Changer operation) | !T |
| !RRS | p1 | **R**un **R**ibbon **S**kip panel & **S**ync |  |
| !CFS |  | Check Feeder Sensor (sensor status) | CF |
| !CRS |  | Check Ribbon Sensor | CR |
| !CRC |  | Check Ribbon Remained Count |  |
| !CSS |  | Check Status and Sensors | S |
| !GVT |  | Get Version/Type (CIT version / Type - 9100, direct) | V |
| !GRS |  | **G**et **R**ibbon **S**ensor value | &RIB |
| !GOI |  | **G**et **O**ption **I**nformation (SAM/MS/IC/RF/Changer장착여부) |  |
| !AOI | p1 | **A**S Set **O**ption **I**nformation (SAM/MS/IC/RF/Changer장착여부 – Diag.) |  |
| !GSI | p1 | **G**et **S**ystem **I**nformation (All Information of p1 –cit/net/scan) |  |
| !ASI | p1 data | **A**S Set **S**ystem **I**nformation (All Information of p1 –cit/net/scan – Diag.) |  |
| !GCV | p1 | Get CIT(printer) parameter Value (each printer parameter) | !V |
| !SCV | p1 p2 | Get CIT(printer) parameter Value (each printer parameter) | !R,+C,… |
| !SRT | p1 p2 | Set Ribbon Type, RF-Tag의 ribbon type response | +RIB |
| !GPC |  | Get Print Count | !V 32 |
| !SPC | p1 | Set Print Count | !SPC |
| !GMA |  | Get MAC Address | +GMA |
| !SMA | p1 | Set MAC Address | +SMA |
| !ASR | p1 p2 p3 | AS/Factory Set RF-TAG Valid No | %RFTAG |
| !ARR |  | AS/Factory Read RF-TAG Valid No | %rftag |
| !RTL |  | Request Terminal Lock |  |
| !RTU |  | Release Terminal Lock |  |
| !SDP | p1 p2 | Set Digital Potentiometer | SLC,SRC,SRS |
| !GDP |  | Get Digital Potentiometer | SGL,SGR, |
| !SDV | p1 | Set Ribbon DC Motor Duty Value | +RMD |
| !GDV |  | Get Ribbon DC Motor Duty Value | !RMD |
| !SPS | p1 | Set Printer Serial Number | SN |
| !GPS |  | S\Get Printer Serial Number | SG |

(“!ASR” 명령은 공장에서 생산 시, “!ARR”은 AS시 사용 가능하다. – 별도의 Code 필요)

: 배포용, TP(TM)에는 “!ASR” 명령은 포함 되지 않아야 한다.

**2) Option Command (&XXX – MS / IC / RF / SAM 등 Option으로 추가되는 기능에 대한 Command)**

M:MS command, I:IC command, R:RF command, **L:Laminator** …

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Data** | **Description** | **CIT** |
| &MEC | p1 | [MS] set Encoder Coercivity | &C |
| &MST | p1 data | [MS] Set(write) Track buffer | &B |
| &MWT | p1 | [MS] Write Track data (Encode Track data) | &E |
| &MRT | p1 | [MS] Read Track data | &L, &L\* |
| &MRJ | p1 | [MS] Read Jitter data after reading the MS data |  |
| &MWJ | p1 | [MS] Read Jitter data after writing and reading the MS data |  |
| &GBT |  | [IC/SAM] Get IFM and SAM B/D type | U |
| &CSI |  | [IC] Check SIM sensor | CSI |
| &CSA |  | [SAM] Check(Get) SAM list | CSA |
| &IMC |  | [IC] Move Smart Card to Docking Station | MS |
| &ISC | data | [IC] Command send to smart card | &IC |
| &SAM | data | [SAM] Send to SAM B/D command | &SAM |
| &PS1 | data | [SAM] Send to PSAM command | &PSAM |
| &PS2 | data | [SAM] Send to PSAM2 command | &psam |
| &RMP |  | [RF] RF card Move contactless Position | MCL |
| &RSC | p1 data | [RF]send RF Command to RF Card command | &RF, &RF2 |
| &ICS | p1 | [IC] change slot number |  |
| &IPO | p1 | [IC] power on |  |
| &ISA | p1 data | [IC] send command |  |
| &IPF | p1 | [IC] power off |  |
| &RMV |  | [RF] Get RF module version |  |
| &RCC | p1 | [RF] Check RF Card |  |
| &RPO | p1 p2 p3 | [RF] Power On |  |
| &RSA | p1 p2 data | [RF] Send Command |  |
| &RPF | p1 | [RF] Power Off |  |
| &RMK | p1 data data | [RF] Set Key (Mifare) |  |
| &RMR | p1 p2 | [RF] Read Data (Mifare) |  |
| &RMW | p1 p2 data | [RF] Write Data (Mifare) |  |
| &RMI | p1 p2 p3 data | [RF] Increment Block (Mifare) |  |
| &RMD | p1 p2 p3 data | [RF] Decrement Block (Mifare) |  |
| &LRP |  | [Laminator] Read Laminator parameters |  |
| &LSP | p1 data | [Laminator] Set Laminator parameters (No write parameter into flash) |  |
| &LWP | p1 data | [Laminator] Write Laminator parameters into flash |  |
| &LMW | p1 data | [Laminator] Set CIT🡨🡪Laminator mode of working |  |
| &LCR |  | [Laminator] Command Status Request – card position, ribbon searched | direct control |
| &LCI |  | [Laminator] Command Initialize Laminator -- Ribbon Sync, Card direction | direct control |
| &LCW |  | [Laminator] Command Warming-Up – Wait Entry card | direct control |
| &LCL |  | [Laminator] Command Laminating (start) | direct control |
| &LCE |  | [Laminator] Command End Laminating | direct control |
| &LCS |  | [Laminator] Command Save mode | direct control |
| &LCJ |  | [Laminator] Command Card Eject |  |
| &LCY |  | [Laminator] Command Ribbon Sync |  |

**3) Print Command ($XXX – 인쇄 관련한 명령어)**

Common printing setting command, Color/Mono/Varnish print command

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Data** | **Description** | **CIT** |
| $SOP | p1 | Start Of Printing |  |
| $STT | p1 | Set On-Time Table | New |
| $SPT | p1 p2 p3 p4 | Set Print Type (Ribbon type, # of print) | +RIB |
| $CCB | p1 p2 | Clear Color/varnish /mono image Buffer | $F, $FP  vF, F, FE |
| $TCB | p1 p2 p3 p4 p5 data | Transfer Color/varnish/mono image Buffer | GT, GS, GH  vG, vZ  G, Z,  GE, EZ |
| $PCP | p1 p2 | Print Color/varnish /mono Print | IS, IV, I, IE |
| $POL | p1 | Print Only Laminator (No printing) |  |
| $CDT | p1 | Calibration Data Transfer (data format에 따름) |  |
| $EOP | p1 | End Of Printing |  |
| $CPJ | P1 | Cancel Printing Job |  |
| $PJS |  | Printing Job Status |  |
| $RPE |  | Re-Print start after Error |  |

**4) SetUp Command (+XXX – 장치 설정 관련한 명령어 = LCD, Memory 설정/확인)**

LCD / Memory data처리

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Data** | **Description** | **CIT** |
| +LMS | p1 p2 data | LCD message save command |  |
| +LDD | p1 p2 data | LCD display Data (data count fixed – 16 characters / line) | +LCD1 |
| +LDO | p1 | LCD display Option command | +LCD2 |
| +LUR | p1 p2 | LCD User’s memory Read (Read from user's LCD memory of flash) | +LCDr  +LCDR |
| +LUW | p1 p2 p3 data | LCD User’s memory Write (Write to user's LCD memory of flash) | +LCDw  +LCDW |
| +UMC | p1 | Clear User Memory | &Kr  &KrE  &KrEE  &Kree |
| +UMR | p1 p2 p3 | Get(Read) User Memory – offset / length 포함 | &KR  &KRE  &KREE  &KRee |
| +UMW | p1 p2 p3 data | Write to user's memory (offset, length, data) | &KW  &KWE  &KWEE  &KWee |

**5) Reference of EEPROM / Status**

**< EEPROM or Flash’s System Memory Structure>**

|  |  |  |  |
| --- | --- | --- | --- |
| CIT Version | Offset 0 | Size 4 | CIT Version |
| Option Info | Offset 4 | Size 4 | Option’s Info.(SAM/RF/MS/RF-TAG/…) |
| CIT parameters (기존정보) | Offset 8 | Size N | TPH/Quality/Color/Mono/MS/IC Parameters |
| RF-Tag/Ribbon Operation. | Offset 8+N | Size M | RF-Tag’s operation method, ribbon info (count) |
| Network Parameter | Offset 8+NM | Size L | Network Values |
| Scanner Parameter | Offset 8+NML | Size K | Scanner Values |
| Customer Info - 1 |  |  | User/Customer’s information for CIT |

**< Status value of Response, 상태 및 Error code >**

|  |  |  |
| --- | --- | --- |
| 분류 | Status Value | 설명 |
| ACK (OK) | 00 00 | ACK – 정상 동작 중 – **명령어 수신 후 동작 중**. (대부분 미 사용) |
|  | 00 06 | ACK – 정상 동작 완료. |
| NACK | 00 15 | NACK – NACK |
| Information | 10 XX | 인쇄 중, MS 동작 중, IC동작 중, ….. |
| Error/Warning | 8X XX | MS / IC / RF / SAM / Ribbon / TPH의 error 정보 |
| Error/Warning | 4X XX | DLL에서 설정하는 오류 코드 - USB 에러, DLL 처리 오류 |

Error clear – Mecha. Error 상태(80 1X)가 아니면, 명령어를 수신하였을 때error를 clear하고, 새로운 명령에 대한 Error와 ACK를 대응한다.

Mecha error는 배출 Key를 눌러 동작 후, Printer Cover를 제거하였을 때 Sensor를 Check하여 clear한다.

# 4. **Command Description**

## **4.1 General Command**

!RRP Command – Run Reset Printer

|  |  |
| --- | --- |
| **Description** | CIT를 Reset한다. / Rebooting 🡪 초기화 🡪 대기 상태 |
| **Syntax** | <ESC>!RRP<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | **NONE** |

!RCE Command – Run Clear Error status

|  |  |
| --- | --- |
| **Description** | Error Code를 Clear한다. – Error 상태에서 복구(모든 Error상태 복구) |
| **Syntax** | <ESC>!RCE<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@RCE<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RBZ Command – Run Buzzer Control

|  |  |
| --- | --- |
| **Description** | Buzzer control On / buzzer Off (p1 = 0) |
| **Syntax** | <ESC>!RBZ<length><data:4bytes><CR> |
| **Parameter** | <length:4> 00000006  <data:2> p1=Retry counts : 0 ~ 511(0x01FF), Off시 0x0000  <data:2> p2=On time (ms): 0 ~ 5000, Off시 0x00  <data:2> p3=Off time (ms): 0 ~ 5000, Off시 0x00 |
| **Response** | <ESC>@RBZ<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RIP Command – Run Input card to Print ready position

|  |  |
| --- | --- |
| **Description** | Card를 Input position인 Print Ready Position으로 이송 시킨다. |
| **Syntax** | <ESC>!RIP<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@RIP<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!REH Command – Run Exit loaded card to output Hopper position

|  |  |
| --- | --- |
| **Description** | Card를 Output Hopper로 배출 시킨다. |
| **Syntax** | <ESC>!REH<length><data><CR> |
| **Parameter** | <length:4> 00000001 |
|  | <Data:1> 0x00 = Normal Eject  0x01 = Hopper가 있을 때, 낱장 배출구로만 카드 배출  0x02 = Hopper가 있을 때, Stacker로만 카드 배출 |
| **Response** | <ESC>@REH<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RTC Command – Run Turn up a Card

|  |  |
| --- | --- |
| **Description** | 카드를 뒤집는다. / Changer장치가 있는 양면 인쇄 기기에서 양면 인쇄 시만 사용  !GOI command의 response(@GOI)를 통해 changer사용 가능여부를 확인 할 수 있다. |
| **Syntax** | <ESC>!RTC<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@RTC<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RRS Command – Run Ribbon Sync & Skip

|  |  |
| --- | --- |
| **Description** | Ribbon Sync. 재 설정한다. Color/Mono Ribbon 관계없이 ribbon sync 다시 설정한다. |
| **Syntax** | <ESC>!RRS<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1> 0x00 = Normal Ribbon (Color/Mono) re-Synchronization, (Color: Next Y panel wait)  0x01 = Only 1 Color Panel skip (Color Ribbon,: Y🡪M🡪C🡪K🡪O) |
| **Response** | <ESC>@RRRS<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RCL Command – Run Cleaning The printer

|  |  |
| --- | --- |
| **Description** | Cleaner Kit으로 Printer 내부를 청소다. |
| **Syntax** | <ESC>!RCL<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@RTC<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RSP Command – Run Self-Printing

|  |  |
| --- | --- |
| **Description** | Self-Printing 동작 한다. Self-Printing Image 인쇄 된다. |
| **Syntax** | <ESC>!RSP<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1> 0x01 = Self-Printing |
| **Response** | <ESC>@RSP<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!RCR Command – Run Card Reject

|  |  |
| --- | --- |
| **Description** | Media 및 카드 Eject 실패 시 Parameter 설정에 따라 카드가 배출을 한다. |
| **Syntax** | <ESC>!RCR<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1> 0x00 = Eject card in front of the printer.  0x01 = Hold the card in front of the printer’s card slot  0x10 = Eject card in the rear printer.  0x11 = Hold the card in the rear printer. |
| **Response** | <ESC>@RCR<status><length><CR>  <Status:2> : ACK/NACK – 동작 완료 후 응답  <length:4>: 00000000 |

!CFS Command – Check Feeder Sensors

|  |  |
| --- | --- |
| **Description** | 단말기 내부의 매체 위치를 체크한다. |
| **Syntax** | <ESC>!CFS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@CFS<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000001  <data:1>:0x00 – 매체 없음 / 0x01-투입구 존재 / 0x02-Stacker 존재 / 0x04-낱장 투입구 존재 / 0x10-내부 존재 / 0x80-IC Contact 되어 있음  => 비트 조합으로 내부 및 투입여부 판단  예) 낱장 투입구와 내부에 카드가 존재 할 때: 0x14  낱장 및 Stacker에 카드 존재: 0x06 |

!CRS Command – Check Ribbon Sensors

|  |  |
| --- | --- |
| **Description** | 현재 장착 된 리본의 종류를 읽는다.  단, 리본종류는 Color(YMCKO, YMCKOK) 와 Mono(K, KO) 로만 전송된다 |
| **Syntax** | <ESC>!CRS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@CRS<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000002  <data:1>: 0x00 – Not Check, 0x30 – Color / 0x34 – Mono (Sensor detect에 의한 결과)  <data:1>: 0x00 – None / 0x31 – YMCKO / 0x32 – YMCKOK /  0x33 – YMC(Half)KO / 0x34 – KO / 0x35 – K  / 0x36 – GOLD/ 0x3b – WHITE/0x3c – SILVER (RF-Tag에 의한 Ribbon 정보)  / 0x3e – RED /0x3f – BLUE (RF-Tag에 의한 Ribbon 정보) |

!CRC Command – Check Ribbon Remained Count

|  |  |
| --- | --- |
| **Description** | 현재 장착 된 리본의 남은 갯수를 읽는다. |
| **Syntax** | <ESC>!CRC<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@CRC<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000002  <data:2>: Ribbon의 남은 갯수 |

!CSS Command – Check Status and Sensors

|  |  |
| --- | --- |
| **Description** | 단말기 현재 상태를 체크한다. |
| **Syntax** | <ESC>!CSS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@CSS<status><length><CR>  <status:2> : Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GVT Command – Get CIT Version / Type

|  |  |
| --- | --- |
| **Description** | 단말기 Firmware의 현재 Version을 확인 한다 |
| **Syntax** | <ESC>!GVT<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GVT<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000008  <data:4>: EEPROM(Flash)’s CIT Version 4 bytes  <data:2>:CIT’s Type (상위 2bytes – Direct: 0000 /Re-transfer: 1010),  <data:2>: CIT’s Model (하위 2bytes – default: 9000 or 9100), 🡪 0x00009100 |

!GRS Command – Get Ribbon Sensor Value

|  |  |
| --- | --- |
| **Description** | 단말기 내 현재 Ribbon의 sensor값을 읽어온다. |
| **Syntax** | <ESC>!GRS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GRS<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000004  <data:4>: Ribbon sensor’s value 4 bytes (현재는 마지막 byte 유효: 00 00 00 ED)  Mecha.구조가 바뀌거나 값이 바뀌면 상위 3 bytes중 사용 가능  상위 2Byte: White LED value, 하위 2Byte:Blue LED value |

!SRS Command – Get Ribbon Sensor Value

|  |  |
| --- | --- |
| **Description** | 단말기 내 Ribbon sensor 값이 적절한 값 (550~650) 이 되도록 White LED의 가변저항 값을 설정한다. |
| **Syntax** | <ESC>!SRS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@SRS<status><length><CR>  <status:2> : Error/ACK  <length:4>: 00000004 |

!GOI Command – Get Option Information Value

|  |  |
| --- | --- |
| **Description** | 단말기 내 Option 정보를 읽어온다. (EEPROM’s Option Info) |
| **Syntax** | <ESC>!GOI<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GOI<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000004  <data:4>: EEPROM(Flash)’s CIT Option 4 bytes (00030000 – MS/IC option장착)  P1: Option 장치 장착 여부 (상위 2bytes)  0001 – MS / **0002 – IC** **/ 0004 – RF1(내장)** / **0008 – RF2(외장)**  0010 – PSAM1 / 0020 – PSAM2 / 0040 – SAM / **0080 -- Hopper**  0100 – RF-Tag(Reader) / 0200 – Changer / **0400 – IFM\_BD\_Type** /  **0800 – SAM\_BD\_Type**  **1000 – Laminator connection (1:Use Laminator, 0: Not use Lmainator)**  **X000– Language (0=한글, 2=영문, 4=중국어, 6=일본어, 8,A,C,E)**  P2: Reserved (하위 2bytes)  \* MS, RF1, RF2, Changer, SAM, PSAM1, PSAM2, LM – 초기화 과정에서 사용여부 확인.  \* RF-Tag(Reader) – “!ASR”명령에 의해서 사용여부 설정됨.  \* IC, Hopper, IFMType, IFMSAMType, Language – “!AOI” 설정 값대로 설정됨.  \* RF1과 RF2는 내.외장 구분이 되지 않으므로 “!AOI”에서 설정해야 함. |

!AOI Command – AS(factory) set Option Information Value

|  |  |
| --- | --- |
| **Description** | 단말기 내 Option 정보를 설정한다. (EEPROM’s Option Info) |
| **Syntax** | <ESC>!AOI<length><data><CR> |
| **Parameter** | <length:4> 00000004  <data:4> EEPROM(Flash)’s CIT Option 4 bytes (ex: 00030000 – MS/IC option장착)  P1: Option 장치 장착 여부 (상위 2bytes)  0001 – MS / **0002 – IC** / **0004 – RF1(내장)** / **0008 – RF2(외장)**  0010 – PSAM1 / 0020 – PSAM2 / 0040 – SAM / **0080 -- Hopper**  0100 – RF-Tag(Reader) / 0200 – Changer // **0400 – IFM\_BD\_Type** /  **0800 – SAM\_BD\_Type**  **1000 – Laminator connection (1:Use Laminator, 0: Not use Lmainator)**  **X000– Language (0=한글, 2=영문, 4=중국어, 6=일본어, 8,A,C,E)**  P2: Reserved (하위 2bytes) |
| **Response** | <ESC>@AOI<status><length><CR>  <status:2> : Error/ACK (Flash/EEPROM writing OK)  <length:4>: 00000000  \* 모든 Field는 Update가 가능하다. 단, 전원 OFF후 POWER-ON시 “MS, RF1, RF2, Changer, SAM, PSAM1 ,PSAM2, RF-Tag(Reader), LM”는 초기화 과정에서 사용여부 재 확인 한다.  \* IC, Hopper, IFM\_BD\_Type, SAM\_BD\_Type, Language – 설정 값대로 설정됨.  \* RF사용시 RF1과 RF2는 내.외장 구분이 되지 않으므로 설정해야 함.  - 0002: IC (1:Use IC module, 0: Not use IC)  **- 0080: Hopper (1:Use Hopper (New hopper only), 0: Not use Hopper)**  **- 0400: IFM BD Type (1:GEMCore PRO, 0:GEMCore Serial)**  **- 0800: SAM BD Type (1:GEMCore PRO, 0: GEMCore Serial)**  **- X000:bit 15,14,13 사용. Language (0=한글, 2=English, 4=中國語, 6=日本語, …., . 까지 사용)**  **값(hex) X000 🡪 (bi) 000x xxxx xxxx xxxx : 한글**  **(bi) 001x xxxx xxxx xxxx : 영어 (English)**  **(bi) 010x xxxx xxxx xxxx : 중국어**  **(bi) 011x xxxx xxxx xxxx : 일본어** |
|  |  |

!GSI Command – Get System Information Value

|  |  |
| --- | --- |
| **Description** | 단말기의 System 정보를 읽어온다. (EEPROM’s Information) |
| **Syntax** | <ESC>!GSI<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=type  0x00– No parameter, 0x01–CIT(Printer) parameter, 0x02– RF-Tag/Ribbon Parameter  0x04 – Network parameter, 0x08 – Scanner parameter, 0x10 – Customer parameter |
| **Response** | <ESC>@GSI<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 000000XX (1+N)  <data:1> Information(parameter) type (0x00, 0x01, 0x02, 0x04, 0x08, 0x10)  <data:N>: EEPROM’s parameter field 내용 – size: N  현재 Size 미 확정 – 진행 중. |

!ASI Command – AS(factory) set System Information Value

|  |  |
| --- | --- |
| **Description** | 단말기의 System 정보를 설정한다. (EEPROM or Flash) |
| **Syntax** | <ESC>!ASI<length><data><CR> |
| **Parameter** | <length:4> 000000XX (1+N)  <data:1>p1=type  0x00– No parameter, 0x01–CIT(printer) parameter, 0x02– RF-Tag/Ribbon Parameter  0x04 – Network parameter, 0x08 – Scanner parameter, 0x10 – Customer parameter  <data:N> EEPROM’s parameter field 내용 – size: N  현재 Size 미 확정 – 진행 중. |
| **Response** | <ESC>@ASI<status><length><CR>  <status:2> : Error/ACK (Flash/EEPROM writing OK)  <length:4>: 00000000 |

!GCV Command – Get CIT(printer) parameter Value (각 항목별)

|  |  |
| --- | --- |
| **Description** | 단말기의 선택된 기존의 CIT Operation parameter 값을 읽어온다. |
| **Syntax** | <ESC>!GCV<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=Printer parameter의 offset |
| **Response** | <ESC>@GCV<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 0000000XX(1+N) – parameter’s offset size에 따름  <data:1> Printer parameter (command 내용)  <data:N> Printer parameter’s Value |

!SCV Command – Set CIT(printer) parameter Value (각 항목별)

|  |  |
| --- | --- |
| **Description** | 단말기의 선택된 기존의 CIT Operation parameter 값을 저장한다. |
| **Syntax** | <ESC>!SCV<length><data><CR> |
| **Parameter** | <length:4> 0000000XX(1+N)  <data:1>p1=Printer parameter의 offset  <data:N>p2=Printer parameter’s Value |
| **Response** | <ESC>@SCV<status><length><data><CR>  <status:2> : Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

[Appendix] CIT(printer) Parameter List (For GCV, SCV command)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Description** |  | **Value** | **Length** |
| 10 | Head resistance | Print Head Label에 표시 되어 있는 Manufacture’s Average Resistance을 설정한다  설정된 Option은 EEPROM에 저장되고, 전원과 관계없이 적용된다. | 2400 ~ 3600, default: 3000 | 2 |
| 11 | Yellow contrast |  |  | 1 |
| 12 | Magenta contrast |  |  | 1 |
| 13 | Cyan contrast |  |  | 1 |
| 14 | Kdye contrast |  |  | 1 |
| 15 | Yellow intensity | Color/Varnish/Monochrome Ribbon Intensity(Heat) Level를 조정한다. | 0 ~ 10, default: 5 | 1 |
| 16 | Magenta intensity |  | 0 ~ 10, default: 5 | 1 |
| 17 | Cyan intensity |  | 0 ~ 10, default: 5 | 1 |
| 18 | Kdye intensity |  | 0 ~ 10, default: 5 | 1 |
| 19 | Black intensity |  | 0 ~ 30, default: 5 | 1 |
| 20 | Varnish intensity |  | 0 ~ 10, default: 5 | 1 |
| 21 | MS Encode Format |  | 0x00=JIS2  0x01=ISO (\*) | 1 |
| 22 | MS Encode Coercivity |  | 0x00=LOCO  0x01=HICO  0x02=AUTO (\*) | 1 |
| 23 | Card input direction | Card의 Input방향을 결정한다. | 0x00=Forward (\*)  0x01=Backward | 1 |
| 24 | Ribbon type | 사용 하는 Ribbon Type을 지정한다. | 0x00=YMCKO (\*)  0x01=KO  0x02=YMCKOK  0x04=Monochrome  0x80=Half YMCKO  0x06 = GOLD  0x0b = White  0x0c = Silver  0x0e = Red  0x0f = Blue | 1 |
| 25 | X-axis print Start position | Offset the Horizontal Start Position | 0 ~ 30, default: 10 | 1 |
| 26 | X-axis print End position | Offset the Horizontal End Position | 0 ~ 20, default: 10 | 1 |
| 27 | Y-axis print Start position | Y-axis Print Start Position (line count ) | 0 ~ 32, deault: 16 | 1 |
| 28 | Pint Value | 인쇄 핀트 조정 값을 설정한다. | 0 ~ 999. default: 112 | 2 |
| 29 | IC contact module status |  |  | 1 |
| 30 | Get Mono print mode | Monochrome Printing Mode(Speed)를 설정한다. | 0x00=Normal Speed + KEE  0x01=Low Speed + KEE  0x04=Normal Speed + KPE  0x05=Low Speed + KPE  0x06=Normal Speed + KPE2  0x07=Low Speed + KPE2 | 1 |
| 31 | Get Card Input Option | Card Input 동작에 대한 옵션을 설정한다.  설정된 Option은 EEPROM에 저장되고, 전원과 관계없이 적용된다. | 0x00=외부에서 대기 (\*)  0x01=내부에서 대기 | 1 |
| 32 | Get Print Count | 인쇄한 매수를 읽어온다. | 0 ~ 100000000, default: 0 | 4 |
| 33 | 리본 Sync 자동 설정 | 리본 Sync 옵션을 설정한다. | 0x00=실행안함 (\*)  0x01=실행 | 1 |
| 34 | 카드 대기 위치 설정 | 인쇄 중 전원을 On/Off을 시켰을 때 카드를 배출하지 않고 대부 대기 위치로 이동한다. | 0x00=동작안함 (\*)  0x01=내부 대기 위치 | 1 |
| 35 | 카드 입력 대기 시간 (sec) | 카드 입력 명령 실행시 카드 입력 대기 시간을 설정한다. | 0 ~ 255, default: 10,  0x00=무한 대기 | 1 |
| 36 | 카드 크기 | Card Size를 설정한다. | 0x00=Normal (ISO7810 – ID1)  0x01=Long Card  0x02=Short Card | 1 |
| 38 | 자동 장애 처리 옵션 | 인쇄 중 장애 발생시 장애 처리 방법에 대한 옵션을 설정한다. | 0 = Auto Option  1.=Manual Option | 1 |
| 39 | Hopper Type의 카드배출 Option | 카드 배출 Option을 설정한다. | 0x00 = Normal  0x01 = 낱장 투입구 배출  0x02 = Stacker 배출  0x03 = 후면배출(Rear) | 1 |
| 40 | 카드 배출 대기 시간(sec) | 카드 배출 후 대기 시간을 설정 한다. | 0 ~ 255, Default 0,  Hopper 없는 Type 적용 | 1 |
| 41 | 리본 Maker 설정 | 리본 Maker를 설정한다. | 0x00 = DPS 0x01 = DNP  0x02 = ITW (New) 0x03 = ITW2 (old) | 1 |
| 42 | Hopper Type의 카드입력 Option | 카드 Insert Option을 설정한다. | 0x00 = Normal  0x01 = Manual Slot Insert | 1 |
| 43 | Dispenser 설정옵션. | 세트의 디스펜서 옵션을 설정한다. | 0x00 : None 0x01: Use |  |

!SRT Command – Set Ribbon Type

|  |  |
| --- | --- |
| **Description** | 사용 하는 Ribbon Type을 지정한다. 이 값은 전원 ON시 초기값으로 사용한다.  인쇄할 때는 인쇄 명령어에 의해 리본 타입이 변경된다. |
| **Syntax** | <ESC>!SRT<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data: 1> p1= Force setting (실제 장착된 ribbon의 종류에 관계없이 강제로 설정)  0x00 = Normal  0x01 = Force setting (RF-Tag를 사용할 경우 강제 설정)  <data: 1> p2=ribbon type 설정 값  0x00 = YMCKO ribbon, 0x01= KreginO, KdyeO ribbon  0x02 = YMCKOK ribbon, 0x04 = K(Mono) ribbon  0x80 = Half YMCKO ribbon , 0x06 = Gold ribbon, 0x0b = White ribbon, 0x0c = Silver ribbon,  0x0e: Red Ribbon, 0x0f: Blue Ribbon |
| **Response** | <ESC>@SRT<status><length><CR>  <status:2> : Error/ACK  Status code value에 따름: Not-matched(Force changed – ribbon type is changed)  <length:4>: 00000000 |

!GPC Command – Get Print Count

|  |  |
| --- | --- |
| **Description** | 인쇄한 매수를 읽어온다. |
| **Syntax** | <ESC>!GPC<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GPC<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000004  <data:4> 인쇄 매수. 0 ~ 100,000,000 |

!SPC Command – Set Print Count (생산/개발에만 사용 가능)

|  |  |
| --- | --- |
| **Description** | 현재까지 인쇄한 매수를 설정한다. (Panel당 1이 증가한다) |
| **Syntax** | <ESC>!SPC<length><data><CR> |
| **Parameter** | <length:4> 00000004  <data:4>p1=인쇄 매수. 0 ~ 100,000,000 |
| **Response** | <ESC>@SPC<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GMA Command – Get MAC Address

|  |  |
| --- | --- |
| **Description** | MAC Address를 읽어온다. |
| **Syntax** | <ESC>!GMA<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GMA<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000006  <data:6>: MAC Address |

!SMA Command – Set MAC Address (생산/개발에만 사용 가능)

|  |  |
| --- | --- |
| **Description** | MAC Address를 설정한다. |
| **Syntax** | <ESC>!SMA<length><data><CR> |
| **Parameter** | <length:4> 00000006  <data:6>p1=MAC Address |
| **Response** | <ESC>@SMA<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!ASR Command – AS/Factory Set RF-TAG Valid No (생산/개발에만 사용 가능)

|  |  |
| --- | --- |
| **Description** | RF-TAG를 사용할 장비(SET)의 Valid-No를 설정한다. |
| **Syntax** | <ESC>!ASR<length><data><CR> |
| **Parameter** | <length:4> 0000000B  <data:2> p1 = 사용여부에 대한 결정  0x7E81 – 미사용  0xA5A5 – 사용  <data: 8> p2= Valid No 8 bytes [0][1][2][3][4][5][6][7]  Valid No중 [2]~[7]이 모두 같으면 명령어 무시.  <data: 1> p3=Check-Sum: p1 + p2 + 0x12 + 0x5C.  – check-sum이 다르면 명령어 무시.  - check-sum 계산 은 XOR(^)를 사용한다. |
| **Response** | <ESC>@ASR<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!ARR Command – AS/Factory Read RF-TAG Valid No

|  |  |
| --- | --- |
| **Description** | RF-TAG를 사용 시 장비(SET)의 Valid-No를 Return한다. |
| **Syntax** | <ESC>!ARR<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@ARR<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000009  <data:8> p1=RF-TAG valid check 결과: 리본 TAG와 valid check 후 OK/FAIL을 나타냄  0x00: OK / 0x01:FAIL (리본 valid check 결과)  - 리본이 없거나, RF-TAG를 사용하지 않으면 FAIL  P2=RF-TAG사용을 위한 Valid-No 8bytes  사용하지 않을 경우 8bytes 전체가 ‘0x00’이다. |

!GMF Command – Get USB Manufacturer

|  |  |
| --- | --- |
| **Description** | USB Manufacturer을 읽어온다. |
| **Syntax** | <ESC>!GMF<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GMF<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 0000000X (최대 32자)  <data:n>: USB Manufacturer |

!SMF Command – Set USB Manufacturer (생산/개발에만 사용 가능)

|  |  |
| --- | --- |
| **Description** | USB Manufacturer을 설정한다. |
| **Syntax** | <ESC>!SMF<length><data><CR> |
| **Parameter** | <length:4> 0000000X (최대 32자)  <data:n>p1: USB Manufacturer |
| **Response** | <ESC>@SMF<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GDS Command – Get USB Description

|  |  |
| --- | --- |
| **Description** | USB Description을 읽어온다. |
| **Syntax** | <ESC>!GDS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GDS<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 0000000X (최대 64자)  <data:n>: USB Description |

!SDS Command – Set USB Description (생산/개발에만 사용 가능)

|  |  |
| --- | --- |
| **Description** | USB Description을 설정한다. |
| **Syntax** | <ESC>!SDS<length><data><CR> |
| **Parameter** | <length:4> 0000000X (최대 64자)  <data:n>p1: USB Description |
| **Response** | <ESC>@SDS<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GSN Command – Get USB Serial Number

|  |  |
| --- | --- |
| **Description** | USB Serial Number을 읽어온다. |
| **Syntax** | <ESC>!GSN<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GSN<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 0000000X (최대 16자)  <data:n>: USB Serial Number |

!SSN Command – Set USB Serial Number (생산/개발에만 사용 가능)

|  |  |
| --- | --- |
| **Description** | USB Serial Number을 설정한다. |
| **Syntax** | <ESC>!SSN<length><data><CR> |
| **Parameter** | <length:4> 0000000X (최대 16자)  <data:n>p1: USB Serial Number |
| **Response** | <ESC>@SSN<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!RTL Command – Request Terminal Lock

|  |  |
| --- | --- |
| **Description** | 단말기를 Lock 상태로 한다. |
| **Syntax** | <ESC>!RTL<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@RTL<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!RTU Command – Release Terminal Lock

|  |  |
| --- | --- |
| **Description** | 단말기 Lock 상태를 해제한다.. |
| **Syntax** | <ESC>!RTU<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@RTU<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!SDP Command – Set Digital Potentiometer

|  |  |
| --- | --- |
| **Description** | 디지털 가변 저항 레벨 값(0~255)을 세팅 하고 저장한다. |
| **Syntax** | <ESC>!SDP<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data: 1> p1= 디지털 가변저항 채널 선택  0x00 = LCD Contrast  0x01 = White LED 밝기 조정  0x02 = Blue LED 밝기 조정  0x04 = White LED 밝기 자동 조정  0x08 = Blue LED 밝기 자동 조정  <data: 1> p2= 디지털 가변 저항 레벨 값 (0~255) |
| **Response** | <ESC>@SDP<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GDP Command – Get Digital Potentiometer

|  |  |
| --- | --- |
| **Description** | 저장된 디지털 가변 저항 레벨 값(0~255)을 불러온다. |
| **Syntax** | <ESC>!GDP<length><CR> |
| **Parameter** | <length:4> 00000001  <data: 1> p1= 디지털 가변저항 채널 선택  0x00 = LCD Contrast  0x01 = White LED 밝기 조정  0x02 = Blue LED 밝기 조정 |
| **Response** | <ESC>@GDP<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000001  <data:1> p1= 디지털 가변 저항 레벨 값(0~255) |

!SDV Command – Set Ribbon DC Motor Duty Value

|  |  |
| --- | --- |
| **Description** | Ribbon DC Motor Duty Value 설정하고 저장한다. |
| **Syntax** | <ESC>!SDV<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data: 1> p1= 30 ~ 80 값을 설정 |
| **Response** | <ESC>@SDV<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GDV Command – Get Ribbon DC Motor Duty Value

|  |  |
| --- | --- |
| **Description** | 저장된 Ribbon DC Motor Duty값을 불러온다. |
| **Syntax** | <ESC>!GDV<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GDV<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000001  <data:1> p1= Ribbon DC Motor Duty값(30 ~ 80) |

!SPS Command – Set Printer Serial Number

|  |  |
| --- | --- |
| **Description** | 장비에 시리얼 번호를 설정한다. |
| **Syntax** | <ESC>!SPS<length>><data><CR> |
| **Parameter** | <length:4> 00000001 ~ 00000016  <data:N> 장비 시리얼 번호 (최대 16자리) |
| **Response** | <ESC>@SPS<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!GPS Command – Get Printer Serial Number

|  |  |
| --- | --- |
| **Description** | 장비에 저장된 시리얼 번호를 불러온다. |
| **Syntax** | <ESC>!GPS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GPS<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000016  <data:16> 장비 시리얼 번호 |

!SLF Command – Set Loader Flag

|  |  |
| --- | --- |
| **Description** | 장비에 Loader Mode 진입하기 위해 Flag를 설정한다. |
| **Syntax** | <ESC>!SLF<length>><data><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@SLF<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!DFW Command – Download FirmWare

|  |  |
| --- | --- |
| **Description** | 장비에 FirmWare Download를 한다. |
| **Syntax** | <ESC>!DFW<length><CR> |
| **Parameter** | <length:4> 0000000X (X+ n)  Data1 : Chksum (N data의 chksum)  DataN : n data (length -1) |
| **Response** | <ESC>@DFW<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!SCB Command – Set Power Control Board

|  |  |
| --- | --- |
| **Description** | CIT에 연결된 Interface Board의 Power Control을 한다. |
| **Syntax** | <ESC>!SCB<length><CR> |
| **Parameter** | <length:4> 00000001  Data1 : 0x01 - Changer On, 0x02 - Changer Off, 0x03 - RF Tag Board On  0x04 - RF Tag Board Off, 0x05 - RF Board On, 0x06 - RF Board Off  0x07 - IFM On, 0x08 - IFM Off, 0x09 - ALL On, 0x0A - ALL Off  0x0B – USB ON, 0x0C – USB OFF |
| **Response** | <ESC>@SCB<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!SCF Command – Savd Configure Data

|  |  |
| --- | --- |
| **Description** | NVM1의 Configure Data를 Parameter영역에 Backup 해준다. |
| **Syntax** | <ESC>!SCF<length><CR> |
| **Parameter** | <length:4> 00000001  Data1 :  0x31 - Configure 1, 사용자가 변경한 값을 저장  Ex) Setup - Advanced Setting – **Save**. [Designer]  0x32 - Configure 2, 공장 초기화 값을 저장  Ex) 장비 속성 설정 [MP]  **※ 디자이너에서는 절대 사용 하지 말 것!**  0x33 - NVM2 |
| **Response** | <ESC>@SCF<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!LCF Command – Load Configure Data

|  |  |
| --- | --- |
| **Description** | Parameter영역의 Data를 NVM1의 영역에 load 해준다. |
| **Syntax** | <ESC>!LCF<length><CR> |
| **Parameter** | <length:4> 00000001  Data1 :  0x31 - Configure 1, 설정되어 있는 값을 읽어옴.  Ex) Setup - Advanced Setting – **Load**. [Designer]  0x32 - Configure 2, 공장 초기화 값을 읽어옴.  Ex) Setup - Advanced Setting – **Default**. [Designer] |
| **Response** | <ESC>@SCB<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!DFE Command – Download Firmware Erase

|  |  |
| --- | --- |
| **Description** | Download 할 Flash Memory 영역을 Erase 해준다. |
| **Syntax** | <ESC>!DFE<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@DFE<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!DFD Command – Download Firmware Data Write

|  |  |
| --- | --- |
| **Description** | Download 할 Flash Memory 영역을 Write 해준다. |
| **Syntax** | <ESC>!DFD<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@DFD<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!DFV Command – Download Firmware Verify

|  |  |
| --- | --- |
| **Description** | Download 할 Flash Memory 의 Data와 받은 Data를 Verify 해준다. |
| **Syntax** | <ESC>!DFV<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@DFV<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!DFR Command – Download Firmware Reset

|  |  |
| --- | --- |
| **Description** | Download 완료 후 장비를 Reset 하여 새로운 APP로 Start 하도록 해준다. |
| **Syntax** | <ESC>!DFR<length><CR> |
| **Parameter** | <length:4> 00000001  Data1 : 0x00 - Success, 0x01 - Fail |
| **Response** | <ESC>@DFR<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!DFS Command – Download Firmware Get Status

|  |  |
| --- | --- |
| **Description** | 진행 상태의 정보를 불러온다. |
| **Syntax** | <ESC>!DFS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@DFS<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000002  <data1> 0x01 - !DFW, 0x02 - !DFE, 0x03 - !DFD, 0x04 - !DFV  <data2> Percent |

!DFF Command – Download Flipper Firmware

|  |  |
| --- | --- |
| **Description** | Flipper에 Firmware를 Download 한다. |
| **Syntax** | <ESC>!DFF<length><data><CR> |
| **Parameter** | <length:4> 000000XX  <data: N> Flipper Firmware |
| **Response** | <ESC>@DFF<status><length><data><CR>  <status:2>: ACK/NACK  <length:4>: 0000000X  <data1> Flipper 응답 Data |

!STD Command – Send To Dispenser

|  |  |
| --- | --- |
| **Description** | Dispenser에 Data를 Bypass하여 전송한다. |
| **Syntax** | <ESC>!STD <length><data><CR> |
| **Parameter** | <length:4> 000000XX  <data: N> Dispenser Packet Data |
| **Response** | <ESC>@STD<status><length><data><CR>  <status:2>: ACK/NACK  <length:4>: 0000000X  <data1> Dispenser 응답 Data |

!EBC Command – Emission Back Card

|  |  |
| --- | --- |
| **Description** | 카드가 체인저를 통해 후면 배출된다.(디스펜서 전용 커맨드) |
| **Syntax** | <ESC>!EBC <length><data><CR> |
| **Parameter** | <length:4> 000000XX  <data: p1> Card Eject Option.  P1 = ‘0’(0x30) – 전면 Nomal 배출.  P1 = ‘1’(0x31) – 낮장 투입구 배출.(미사용)  P1 = ‘2’(0x32) – Stacker 배출.(미사용)  P1 = ‘4’(0x34) – 후면 카드 배출하여 후면 Roller에 물린 상태  P1= ‘5’(0x35) – 정면 카드 배출되면서 전면에 물리지 않은 상태, 밖으로 배출  P1= ‘6’(0x36) – 후면 카드 배출되면서 후면 Roller에 물리지 않은 상태, 밖으로 배출  (정방향의 경우 Reject)  P1= ‘8’(0x38) –Flipper를 회전 배출하기 위한 회전 Paramter, 밖으로 배출 |
| **Response** | <ESC>@EBC<status><length><data><CR>  <status:2>: ACK/NACK  <length:4>: 0000000X |

!SIA Command – Set IP Address

|  |  |
| --- | --- |
| **Description** | IP 관련 정보를 설정한다. |
| **Syntax** | <ESC>!SIA<length><data><CR> |
| **Parameter** | <length:4> 00000005  <data1> p1= 네트워크 정보 인덱스  0x00 = IP address  0x01 = Subnet mask  0x02 = Gateway  <data4> 네트워크 정보 인덱스에 해당하는 값 (IP address 또는 Subnet mask 또는 Gateway) |
| **Response** | <ESC>@SDP<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |
|  |  |

!GIA Command – Get IP Address

|  |  |
| --- | --- |
| **Description** | IP 관련 정보를 불러온다. |
| **Syntax** | <ESC>!GIA<length><CR> |
| **Parameter** | <length:4> 00000001  <data: 1> p1= 네트워크 정보 인덱스  0x00 = IP address  0x01 = Subnet mask  0x02 = Gateway |
| **Response** | <ESC>@GDP<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000004  <data:4> 네트워크 정보 인덱스에 해당하는 값 (IP address 또는 Subnet mask 또는 Gateway) |
|  |  |
|  |  |

!SSC Command – Set Ribbon Sensor Change Flag

|  |  |
| --- | --- |
| **Description** | Ribbon Sensor의 위치 관련 정보를 설정한다. |
| **Syntax** | <ESC>!SSC<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data1> 0x00 = Original Ribbon Sensor 위치  0x01 = 변경된 Ribbon Sensor 위치 |
| **Response** | <ESC>@SSC<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |
|  |  |

!GSC Command – Get Ribbon Sensor Change Flag

|  |  |
| --- | --- |
| **Description** | Ribbon Sensor의 위치 관련 정보를 불러온다. |
| **Syntax** | <ESC>!GSC<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GSC<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000001  <data1> 0x00 = Original Ribbon Sensor 위치  0x01 = 변경된 Ribbon Sensor 위치 |

!SRR Command – Set RF-TAG Ribbon Type

|  |  |
| --- | --- |
| **Description** | RF-TAG의 Ribbon Type별로 사용을 설정한다. |
| **Syntax** | <ESC>!SRR<length><CR> |
| **Parameter** | <length:4> 00000001  <data1> 0x00 - Color, Mono 설정- , 0x01 - Color만 설정,  0x02 - Mono만 설정, 0x03 - Color, Mono 둘 다 설정 안됨 |
| **Response** | <ESC>@SRR<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |
|  |  |

!GRR Command – Get RF-TAG Ribbon Type

|  |  |
| --- | --- |
| **Description** | RF-TAG의 설정된 Ribbon Type 정보를 불러온다. |
| **Syntax** | <ESC>!GRR<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@GRR<status><length><data><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000001  <data1> 0x00 - Color, Mono 설정- , 0x01 - Color만 설정,  0x02 - Mono만 설정, 0x03 - Color, Mono 둘 다 설정 안됨 |

!LRM Command – Load Ribbon Maker (New Command)

|  |  |
| --- | --- |
| **Description** | EEPROM의 리본 Maker값을 읽어 Flash영역에 저장한다 |
| **Syntax** | <ESC>!LRM<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@LRM<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX(N)  <data:N>p1: Ribbon maker (0x00:DPS 0x01:DNP 0x02:ITW 0x03:ITW2) |

!SRM Command – Set Ribbon Maker (New Command)

|  |  |
| --- | --- |
| **Description** | Ribbon Maker 종류를 Flash와 EEPROM에 저장/세팅한다. |
| **Syntax** | <ESC>!SRM<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:6>p1= Ribbon maker (0x00:DPS 0x01:DNP 0x02:ITW 0x03:ITW2) |
| **Response** | <ESC>@SRM<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!SEI Command – Set EEPROM Init (New Command)

|  |  |
| --- | --- |
| **Description** | Ribbon Maker 종류를 Flash와 EEPROM에 저장/세팅한다. |
| **Syntax** | <ESC>!SEI<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data1> 0x01 - EEPROM Count Init (8M Full Download전 반드시 실행)  0x02 – EEPROM Init Flag, (8M Full Download시 EEPROM 사용 Flag Init)  0x03 – Factory Default data (eeprom, nvm)  <data2> data1의 0x01,0x03 – don’t care data 1의 0x02 – data 값(0xff-eeprom data 초기화, 0xff가아닌값은 데이터 유지) |
| **Response** | <ESC>@SEI<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

!PSS Command – Printer Sensor Status(Only Kiosk 향)

|  |  |
| --- | --- |
| **Description** | Kiosk 향의 카드의 센서 위치 및 Ready 상태 여부 확인한다. |
| **Syntax** | <ESC>!PSS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>@PSS<status><length><data><CR>  <status:2> : Error/ACK  <length:4>: 00000001  <data:1>:0x00 – 매체 없음 / 0x01-Path Sensor 1번에 존재 / 0x02-Path Sensor 2번에 존재 / 0x04-Path Sensor 3번에 존재 / 0x08-Flipper 내부에 존재 / 0x80-카드가 Ready 위치 여부 확인  => 비트 조합으로 내부 및 투입여부 판단  \* Flipper 반대 편인 전면부의 센서를 1번으로 가운데 센서가 2번 끝에 있는 센서가 3번 센서  Ex) 카드가 2번 센서에서 Ready 위치에 있을 경우: 0x82 |

## **4.2 Option Command**

&MEC Command – MS set Encoder Coercivity

|  |  |
| --- | --- |
| **Description** | Magnetic encoder의 Coercivity를 지정한다. |
| **Syntax** | <ESC>&MEC<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1> 0x00–Low(Lo-Co), 0x01-High(Hi-Co), 0x02-Auto |
| **Response** | <ESC>\*MEC<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&MST Command – MS Set(write) Track buffer

|  |  |
| --- | --- |
| **Description** | Encoding될 Track의 Data buffer에 Data를 loading한다. Encoding명령 전에 이 명령으로 Encoding Data를 우선적으로 Loading해야 한다. |
| **Syntax** | <ESC>&MST<length><data><CR> |
| **Parameter** | <length:4> 000000XX (1+N)  <data: 1>p1=track number.  0x00: No parameter, 0x01: Track1, 0x02: Track2, 0x04: Track3, 0x08: JIS-2, 0x0F: All Track  <data: N> 각 Track의 decimal data string  ex) abcdefghij23 🡪 61h 62h 63h 64h 65h 66h 67h 68h 69h 6Ah 32h 33h |
| **Response** | <ESC>\*MST<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&MWT Command – MS Write Track data

|  |  |
| --- | --- |
| **Description** | 지정된 Track을 Write한다. (Verify포함) |
| **Syntax** | <ESC>&MWT<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=track number. - 0x01: Track1, 0x02: Track2, 0x04: Track3, 0x08: JIS-2, 0x0F: All Track |
| **Response** | <ESC>\*MWT<status><length><data><CR>  <status:2> : Error/ACK (MS status)  <length:4>: 00000004  <data:1>: Command’s Track data (0x01, 0x02, 0x04, 0x07, …)  <data:1>: OK/FAIL : 0x00 – No Error (Verify OK) / 0x01 – Track Verify Error  <data:2>: Error rate, Jitter등 information : default = 0x0000 |

&MRT Command – MS Read Track data

|  |  |
| --- | --- |
| **Description** | Magnetic Card의 지정된 Track으로부터 Data를 읽는다. |
| **Syntax** | <ESC>&MRT<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=track number  0x01: Track1, 0x02: Track2, 0x04: Track3, 0x08: JIS-2, 0x0F: All Track |
| **Response** | <ESC>\*MRT<status><length><data><CR>  <status:2> : Error/ACK (Status)  <length:4>: 000000XX (1+N) or 0000XXXX (3+N)  **Command의 p1(track number) = 0x01, 0x02, 0x04 (각 Track일 때):**  <data:1>: Command’s Track number (0x01, 0x02, 0x04)  <data: N> 해당 Track의 ASCII data string  ex) abcdefghij23 🡪 61h 62h 63h 64h 65h 66h 67h 68h 69h 6Ah 32h 33h  **Command의 p1(track number) = 0x07(All track일 때):**  <data:1> 1 Track number (0x11)  <data: N> 1 Track의 ASCII data string  <data:1> 2 Track number (0x12)  <data: N> 2 Track의 ASCII data string  <data:1> 3 Track number (0x13)  <data: N> 3 Track의 ASCII data string  <data:1> JIS-2 number (0x14)  <data: N> JIS-2 의 ASCII data string  ex) track1 data = abc , track2 data = 1234, track3 data = 6789  🡪 0x01,0x61,0x62,0x63,0x02,0x31,0x32,0x33,0x34,0x04,0x66,0x67,0x68,0x69  ex) track1 data = 4567, track2 data = data 없음, track3 data = 6789  🡪 0x01, 0x34,0x35,0x36,0x37,0x04,0x64,0x65,0x66,0x67 |

&GBT Command – Get IFM and SAM B/D type (Check GemCore Serial and Pro)

|  |  |
| --- | --- |
| **Description** | Get IFM/SAM B/D type. |
| **Syntax** | <ESC>&GBT<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*GBT<status><length><data><CR>  <status:2> : Error/ACK (Status)  <length:4>: 00000001 – IFM status’s 상태 길이  <data: 1> Board status values – 0x30, 0x31, 0x32, 0x33  Bit 0: 0: IFM GemCore Serial, 1: IFM GemCore POS Pro  Bit 1: 0: SAM GemCore Serial, 1: SAM GemCore POS Pro |

&CSI Command – Check SIM sensor

|  |  |
| --- | --- |
| **Description** | Check SIM sensor. |
| **Syntax** | <ESC>&CSI<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*CSI<status><length><data><CR>  <status:2> : Error/ACK (Status)  <length:4>: 00000001 – SIM sensor의 상태 길이  <data: 1> SIM Sensor status – ‘0’:None, ‘1’:Exist |

&CSA Command – Check SAM list

|  |  |
| --- | --- |
| **Description** | SAM Board의 각 Socket에 SAM Chip이 있는 소켓 번호를 읽어온다. |
| **Syntax** | <ESC>&CSA<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*CSA<status><length><data><CR>  <status:2> : Error/ACK (Status)  <length:4>: 00000001 – SAM List’s 길이  <data: 0> SAM칩 장착된 소켓번호 정보.  bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0  ┌────┬────┬────┬────┬────┬────┬────┬────┐  │ 0 │SAM7│SAM6│SAM5│SAM4│SAM3│SAM2│SAM1│  └────┴────┴────┴────┴────┴────┴────┴────┘  해당 비트값이 1이면 칩이 장착된 상태임.  예) Sam chip이 1, 3에 있을때 : pStatus = 05h |

&IMC Command – IC Move Smart Card to Docking Station

|  |  |
| --- | --- |
| **Description** | Card를 Smart Card 위치로 이송 시킨다. |
| **Syntax** | <ESC>&IMC<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*IMC<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&ISC Command – IC Send Command to Smart Card

|  |  |
| --- | --- |
| **Description** | IC command를 전송한다. |
| **Syntax** | <ESC>&ISC<length><data><CR> |
| **Parameter** | <length:4> 0000XXXX (N)  <data: N> IC/SAM Command Format |
| **Response** | <ESC>\*ISC<status><length><CR>  <status:2> Error/ACK  <length:4> 0000000X (N)  <data:n> IC/SAM Response Format |

&SAM Command – SAM Send Command

|  |  |
| --- | --- |
| **Description** | SAM command를 전송한다.. |
| **Syntax** | <ESC>&SAM<length><data><CR> |
| **Parameter** | <length:4> 000000XX (N)  <data: N> IC/SAM Command Format |
| **Response** | <ESC>\*SAM<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> IC/SAM Response Format |

&PS1 Command – PSAM Send Command

|  |  |
| --- | --- |
| **Description** | PSAM command를 전송한다.. |
| **Syntax** | <ESC>&PS1<length><data><CR> |
| **Parameter** | <length:4> 000000XX (N)  <data: N> PSAM Command Format |
| **Response** | <ESC>\*PS1<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> PSAM Response Format |

&PS2 Command – PSAM2 Send Command

|  |  |
| --- | --- |
| **Description** | PSAM2 command를 전송한다.. |
| **Syntax** | <ESC>&PS2<length><data><CR> |
| **Parameter** | <length:4> 000000XX (N)  <data: N> PSAM Command Format |
| **Response** | <ESC>\*PS2<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> PSAM Response Format |

&RMP Command – RF Card MoveContactless Position

|  |  |
| --- | --- |
| **Description** | RF card를 contactless B/D위치로 이송 시킨다. |
| **Syntax** | <ESC>&RMP<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*RMP<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&RSC Command – RF Send Command to RF card

|  |  |
| --- | --- |
| **Description** | RF command를 전송한다. (내장/외장) |
| **Syntax** | <ESC>&RSC<length><data><CR> |
| **Parameter** | <length:4> 0000XXXX (1+N)  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈)  <data: N> RF Command Format |
| **Response** | <ESC>\*RSC<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> RF Response Format |

[Appendix] IC/SAM Command/Response Format

**1) Command Format (IFM Board – GEMCORE SERIAL)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 2 | 2 | N\*2 | 2 | 1 |
| STX | Length | Command | Data | BCC | ETX |
| 60h | 1+N |  |  | xxh | 03h |

\* Length: Command, Data

\* BCC: Exclusive OR(STX, Length, Command, Data )

**2) Response Format (IFM Board – GEMCORE SERIAL)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 2 | 2 | N\*2 | 2 | 1 |
| STX | Length | Status | Response | BCC | ETX |
| 60h | 1+N |  |  | xxh | 03h |

\* Length: Status, Response

\* BCC: Exclusive OR (STX, Length, Status, Response)

**3) Command/Response Format (IFM Board – GEMCORE POS PRO)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 10 | | | | | | | Length | 1 |
| 1 | 4 | 1 | 1 | 1 | 1 | 1 |
| Message | Length | Slot | Sequence | bRFU1 | bRFU2 | bRFU3 |
| SYNC | CTRL | CCID\_HDR | | | | | | | [DATA] | LRC |
| 03h | 06h |  | | | | | | | .. | xxh |

\* Length: Length of DATA.

\* LRC: Exclusive OR (SYNC, CTRL, CCID\_HDR, DATA)

[Appendix] PSAM Command/Response Format

**1) Command/Response Format**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | n | 1 | 2 |
| STX | Length | Data | ETX | Checksum |
| 02h | 3+n |  | 03h | xxh xxh |

\* Length: Data

[Appendix] RF Command/Response Format

**1) Command Format**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 1 | n | 1 | 1 |
| STX | Length | Command | Data | BCC | ETX |
| 02h | 2+n |  |  | xxh | 03h |

\* Length: Command, Data, BCC

\* BCC: Exclusive OR (Length, Command, Data)

**2) Response Format**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 1 | n | 1 | 1 |
| STX | Length | Status | Response | BCC | ETX |
| 02h | 2+n |  |  | xxh | 03h |

\* Length: Status, Response, BCC

\* BCC: Exclusive OR (Length, Status, Response)

&ICS Command – IC Change Slot Number (New Command)

|  |  |
| --- | --- |
| **Description** | [IC] 슬롯 번호 변경 |
| **Syntax** | <ESC>&ICS<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=slot number – 1:Mobile, 1~5:SAM Aux Slot |
| **Response** | <ESC>\*ICS<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&IPO Command – IC Power On (New Command)

|  |  |
| --- | --- |
| **Description** | [IC] Power On  Aux Slot인 경우에는 slot을 변경한다. |
| **Syntax** | <ESC>&IPO<length><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=slot number |
| **Response** | <ESC>\*IPO<status><length><data><CR>  <status:2> Error/ACK  <length:4> 00000XXX(N)  <data:N>p1=ATR |

&ISA Command – IC Send Command (New Command)

|  |  |
| --- | --- |
| **Description** | [IC] APDU 전송 |
| **Syntax** | <ESC>&ISA<length><data><CR> |
| **Parameter** | <length:4> 000000XX (N)  <data:1>p1=slot number  <data:N>APDU |
| **Response** | <ESC>\*ISA<status><length><data><CR>  <status:2> Error/ACK  <length:4> 00000XXX(N)  <data:N>p1=Response data |

&IPF Command – IC Power oFf (New Command)

|  |  |
| --- | --- |
| **Description** | [IC] Power Off |
| **Syntax** | <ESC>&IPF<length><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=slot number |
| **Response** | <ESC>\*IPF<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&MRJ Command – read Jitter data after Reading MS data

|  |  |
| --- | --- |
| **Description** | MS Read 후 MS Jitter data를 보낸다. |
| **Syntax** | <ESC>&MRJ<length><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=ms track select (0x01:Track1, 0x02:Track2, 0x04:Track3) |
| **Response** | <ESC>\*MRJ<status><length><data><CR>  <status:2> Error/ACK  <length:4> xxxxxxxx  <data:n>Threshold+4byte ascii jitter data+0x20+4byte ascii jitter data+…+4byte ascii jitter data |
|  |  |

&MWJ Command – read Jitter data after Writing and reading MS data

|  |  |
| --- | --- |
| **Description** | MS Write and read 후 MS Jitter data를 보낸다. |
| **Syntax** | <ESC>&MWJ<length><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=ms track select (0x01:Track1, 0x02:Track2, 0x04:Track3) |
| **Response** | <ESC>\*MWJ<status><length><data><CR>  <status:2> Error/ACK  <length:4>xxxxxxxx  <data:n>Threshold+4byte ascii jitter data+0x20+4byte ascii jitter data+…+4byte ascii jitter data |

&RMV Command – RF Get Module Version (New Command)

|  |  |
| --- | --- |
| **Description** | [RF] RF 모듈 버전 읽어오기 |
| **Syntax** | <ESC>&RMV<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*RMV<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX(N)  <data:N>p1: RF module version data |

&RCC Command – RF Check Card (New Command)

|  |  |
| --- | --- |
| **Description** | RF 카드 중복 체크 |
| **Syntax** | <ESC>&RCC<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈) |
| **Response** | <ESC>\*RCC<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> RF module response data |

&RPO Command – RF Power On (New Command)

|  |  |
| --- | --- |
| **Description** | RF 카드 Power On |
| **Syntax** | <ESC>&RPO<length><data><CR> |
| **Parameter** | <length:4> 00000003  <data:1>p1=RF Card Type(0:Mifare, 1:15693, 2:14443 Type A, 3: 14443 Type B)  <data:1>p2=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:1>p3=Time (100 ms) |
| **Response** | <ESC>\*RPO<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> RF module response data |

&RSA Command – RF Send Command (New Command)

|  |  |
| --- | --- |
| **Description** | RF 카드에 APDU를 보낸다. |
| **Syntax** | <ESC>&RSA<length><data><CR> |
| **Parameter** | <length:4> 000000XX(2+N)  <data:1>p1=RF Card Type(1:15693, 2:14443 Type A, 3: 14443 Type B)  <data:1>p2=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:n>p3=APDU |
| **Response** | <ESC>\*RSA<status><length><data><CR>  <status:2> Error/ACK  <length:4> 000000XX (N)  <data:N> RF module response data |

&RPF Command – RF Power Off (New Command)

|  |  |
| --- | --- |
| **Description** | RF 카드 Power Off |
| **Syntax** | <ESC>&RPO<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:1>p2=Time (0 ms) |
| **Response** | <ESC>\*RPF<status><length><data><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&RMK Command – RF Set Key (Mifare) (New Command)

|  |  |
| --- | --- |
| **Description** | Mifare Key 설정 |
| **Syntax** | <ESC>&RMK<length><data><CR> |
| **Parameter** | <length:4> 00000007  <data:1>p1=Key Type (‘A’, ‘B’)  <data:2>p2=Block ( sector \* 4 + block)  <data:6>p3=Key A(B) DATA |
| **Response** | <ESC>\*RMK<status><length><data><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&RMR Command – RF Read Data (Mifare) (New Command)

|  |  |
| --- | --- |
| **Description** | Mifare Data Read |
| **Syntax** | <ESC>&RMR<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:1>p2=Block ( sector \* 4 + block) |
| **Response** | <ESC>\*RMR<status><length><data><CR>  <status:2> Error/ACK  <length:4> 00000016  <data:16> Mifare data |

&RMW Command – RF Write Data (Mifare) (New Command)

|  |  |
| --- | --- |
| **Description** | Mifare Data Write |
| **Syntax** | <ESC>&RMW<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:1>p2=Block ( sector \* 4 + block)  <data:16>p3=Mifare data |
| **Response** | <ESC>\*RMW<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&RMI Command – RF Increment Block (Mifare) (New Command)

|  |  |
| --- | --- |
|  | |
| **Description** | Source Block 값을 입력 값만큼 증가시켜 Target Block에 저장한다. (**Mifare 전용 함수**) |
| **Syntax** | <ESC>&RMI<length><data><CR> |
| **Parameter** | <length:4> 00000007  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:1>p2=카드 잔액이 들어있는 블록  <data:1>p3=증가된 값이 들어갈 블록  <data:4>p4=증가 값 |
| **Response** | <ESC>\*RMI<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

&RMD Command – RF Decerement Block (Mifare) (New Command)

|  |  |
| --- | --- |
|  | |
| **Description** | Source Block 값을 입력 값만큼 감소시켜 Target Block에 저장한다. (**Mifare 전용 함수**) |
| **Syntax** | <ESC>&RMD<length><data><CR> |
| **Parameter** | <length:4> 00000007  <data:1>p1=Antenna select (0x01:내부 RF 모듈, 0x02: 외부 RF 모듈, 0x03: 내부/외부 RF 모듈)  <data:1>p2=카드 잔액이 들어있는 블록  <data:1>p3=감소된 값이 들어갈 블록  <data:4>p4=감소 값 |
| **Response** | <ESC>\*RMD<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

## **4.3 Laminator Command**

&LRP Command – Laminator Read Paramters (New Command)

|  |  |
| --- | --- |
| **Description** | Laminator의 parameter 값을 읽어 온다. |
| **Syntax** | <ESC>&LRP<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*LRP<status><length><data><CR>  <status:2> Error/ACK  <length:4> 0000009  <data:1> Pass Mode: 0-Auto(default), 1:LM Up, 2:LM Down, 3: Both, 4: by-pass  <data:2> Film Type: 0-Overlay, 1:patch-type1 (default), 2: patch-type2  <data:3> Control: 0-Auto(default), 1-Manual  <data:4> Temperature: 150~195도 10단계 (150, 155, 160, 165, …… 195)  <data:5> Speed:18mm/sec (default),  <data:6> FAN: 0- fan-full (default), 1- fan-half, 2-fan-off  <data: 7> CoolTime: 30- default, 5~30: 5초 단위 값  <data:8> PowerSave: 0-Use power save, 1-No use power save  <data:9> Direction: Lamination후 card 이송 방향, 0-forward, 1-backward |

&LSP Command – Laminator Set Paramters (No write into flah)

|  |  |
| --- | --- |
| **Description** | Laminator의 parameter 값을 Laminator에 전달한다. |
| **Syntax** | <ESC>&LSP<length><data><CR> |
| **Parameter** | <length:4> 00000009  <data:1> Pass Mode: 0-Auto(default), 1:LM Up, 2:LM Down, 3: Both, 4: by-pass  <data:2> Film Type: 0-Overlay, 1:patch-type1 (default), 2: patch-type2  <data:3> Control: 0-Auto(default), 1-Manual  <data:4> Temperature: 150~195도 10단계 (150, 155, 160, 165, …… 195)  <data:5> Speed:18mm/sec (default),  <data:6> FAN: 0- fan-full (default), 1- fan-half, 2-fan-off  <data: 7> CoolTime: 30- default, 5~30: 5초 단위 값  <data:8> PowerSave: 0-Use power save, 1-No use power save  <data:9> Direction: Lamination후 card 이송 방향, 0-forward, 1-backward |
| **Response** | <ESC>\*LSP<status><length><CR>  <status:2> Error/ACK  <length:4> 0000000 |

&LWP Command – Laminator Write Paramters (Write into flah)

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator의 parameter 값을 Laminator의 Flash에 write한다. |
| **Syntax** | <ESC>&LSP<length><data><CR> |
| **Parameter** | <length:4> 00000009  <data:1> Pass Mode: 0-Auto(default), 1:LM Up, 2:LM Down, 3: Both, 4: by-pass  <data:2> Film Type: 0-Overlay, 1:patch-type1 (default), 2: patch-type2  <data:3> Control: 0-Auto(default), 1-Manual  <data:4> Temperature: 150~195도 10단계 (150, 155, 160, 165, …… 195)  <data:5> Speed:18mm/sec (default),  <data:6> FAN: 0- fan-full (default), 1- fan-half, 2-fan-off  <data: 7> CoolTime: 30- default, 5~30: 5초 단위 값  <data:8> PowerSave: 0-Use power save, 1-No use power save  <data:9> Direction: Lamination후 card 이송 방향, 0-forward, 1-backward |
| **Response** | <ESC>\*LSP<status><length><CR>  <status:2> Error/ACK  <length:4> 0000000 |

&LMW Command – Laminator CIT 🡨🡪 Laminator interface working mode

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator와 interface를 설정한다. 동작 방법에 따라 명령어 처리가 달라진다. |
| **Syntax** | <ESC>&LMW<length><data><CR> |
| **Parameter** | <length:4> 00000004  <data:1> PC/CIT/Laminator가 각각 동작에 대한 우선권을 설정한다.  0: 동작 관련하여, CIT와 Laminator가 우선권을 쥐고 판단한다. (default)  1: PC에서 Laminator를 제어한다.  <data:2~4> Reseved-1~3 |
| **Response** | <ESC>\*LMW<status><length><CR>  <status:2> Error/ACK  <length:4> 0000000 |

&LCR Command – Laminator command: Status Get Request

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator의 상태를 읽어온다. |
| **Syntax** | <ESC>&LCR<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*LCR<status><length><data><CR>  <status:2> Error/ACK  <length:4> 0000002  <data:1> Path Sensor : senor On-Off info  Ph4-ph3-ph2-ph1 : b3-b2-b1-b0  Cover sensor: b4  Heater Up/Down sensor: b5  **Heater Not Ready (Over-heat/No-working-heat): b6**  **Heater Ready (Ready Warming-Up Temp): b7**  <data:2> Ribbon Sensor: Detected & current (white/black)  Ribbon searched – UP : b0  Ribbon searched – DOWN : b1  Up detection – Black/White : b4 (1/0)  Down detection – Black/White : b5 (1/0) |

&LCI Command – Laminator command: Initialize Request

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator Initialize. |
| **Syntax** | <ESC>&LCI<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data:1> Card move  0: Eject card Auto detection  1: Eject card to front  2: Eject card to back  3: No move card  <data:2> Ribbon Sync (search)  0: Use PATH mode of Laminator  1: Search Up ribbon  2: Search Down ribbon  3: Search Both ribbon |
| **Response** | <ESC>\*LCI<status><length><data><CR>  <status:2> Error/ACK  <length:4> 0000002  <data:1> Card moved  0:Card JAM 1: Card ejected front, 2: Card ejected back, 3:No moved  <data:2> Ribbon sync result - Detected & current (white/black)  Ribbon searched – UP : b0  Ribbon searched – DOWN : b1  Up detection – Black/White : b4 (1/0)  Down detection – Black/White : b5 (1/0) |

&LCW Command – Laminator command: Warming-Up (Card entry wait)

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator Warming-Up & wait card entry. |
| **Syntax** | <ESC>&LCW<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data:1> Waiting time – card entry  <data:2> Reserved |
| **Response** | <ESC>\*LCW<status><length><data><CR>  <status:2> Error/ACK  <length:4> 0000002  <data:1>  <data:2> |

&LCL Command – Laminator command: Laminating Proceess

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator working run. |
| **Syntax** | <ESC>&LCL<length<CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*LCL<status><length><CR>  <status:2> Error/ACK (ACK is started laminator)  <length:4> 0000000 |

&LCE Command – Laminator command: Laminating End Proceess

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator working stop. |
| **Syntax** | <ESC>&LCE<length<CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*LCE<status><length><CR>  <status:2> Error/ACK  <length:4> 0000000 |

&LCS Command – Laminator command: Laminator Save Mode

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator working save mode enter. |
| **Syntax** | <ESC>&LCS<length<CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>\*LCS<status><length><CR>  <status:2> Error/ACK  <length:4> 0000000 |

&LCJ Command – Laminator command: Card Eject

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator card eject. |
| **Syntax** | <ESC>&LCJ<length<CR> |
| **Parameter** | <length:4> 00000001  <data:1> Card move  0: Eject card Auto detection (default)  1: Eject card to front  2: Eject card to back  3: No move card |
| **Response** | <ESC>\*LCJ<status><length><CR>  <status:2> Error/ACK  <length:4> 0000001  <data:1> Card Moved  0:Card JAM 1: Card ejected front, 2: Card ejected back, 3:No moved |

&LCY Command – Laminator command: Laminator Ribbon Sync

|  |  |
| --- | --- |
|  | |
| **Description** | Laminator command – Laminator ribbon sync. |
| **Syntax** | <ESC>&LCY<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1> 0: Use PATH mode of Laminator  1: Search Up ribbon  2: Search Down ribbon  3: Search Both ribbon |
| **Response** | <ESC>\*LCY<status><length><data><CR>  <status:2> Error/ACK  <length:4> 0000001  <data:1> Ribbon sync result - Detected & current (white/black)  Ribbon searched – UP : b0  Ribbon searched – DOWN : b1  Up detection – Black/White : b4 (1/0)  Down detection – Black/White : b5 (1/0) |

## **4.4 Print Command**

**컬러 리본으로 인쇄시 명령어 전송 순서:**

1) 이미지 저장 모드 설정(옵션): $IFO 0x00 🡪

2) Y-axis value를 읽어온다: !GCV 🡪 @GCV (🡨)

3) 인쇄 타입 설정: $SPT 0x00 0x01 🡪

4) 버퍼 초기화: $CCB 0x1F 0x00 🡪

5) 데이터 전송: $TCB 0x01 🡪 $TCB 0x02 🡪 $TCB 0x04 🡪 $TCB 0x08 🡪 $TCB 0x10 🡪

6) 보정 데이터 전송: $CDT 🡪

7) 인쇄: $PCP 0x00 (또는 $PCP 0x01 🡪 $PCP 0x02 🡪 $PCP 0x04 🡪 $PCP 0x08 🡪 $PCP 0x10) 🡪

8) 카드 배출(옵션): !REH

**모노(K) 리본으로 인쇄시 명령어 전송 순서:**

1) 이미지 저장 모드 설정(옵션): $IF O 0x00 🡪

2) Y-axis value를 읽어온다: !GCV 🡪 @GCV (🡨)

3) 인쇄 타입 설정: $SPT 0x04 0x01 🡪

4) 버퍼 초기화: $CCB 0x20 0x00 🡪

5) 데이터 전송: $TCB 0x20 🡪

6) 인쇄: $PCP 0x00 (또는 $PCP 0x20) 🡪

7) 카드 배출(옵션): !REH

STT Command는 매 인쇄 시 마다 사용할 필요는 없다.

“STT” 명령어는 On-Time 및 Data-Bit 순서를 바꿀 때 사용한다.

“STT” 명령어를 사용하지 않을 때는 Default값을 사용한다. (Default On-Time과 Data순서는 명령어에서 확인)

일반적으로 Booting 후 첫 인쇄 시나, Booting후 초기화 과정에서 한번 적용하면 Power-Off전까지는 그 값을 그대로 사용한다.

$STT Command – Set On-Time Table

|  |  |
| --- | --- |
| **Description** | Set에 pixel의 data(bit) format과 그에 해당하는 On-Time값을 설정한다.  각 Panel별 On-Time rate를 달리 줄 수 있도록, color별 On-Time table를 사용한다. |
| **Syntax** | <ESC>$STT<length><data><CR> |
| **Parameter** | <length:4> 0x00000116 (278 = 64+64+64+64+8+3+8+3)  <data:1>p1 = On-Time + Data format (Color )  On-Time + Bits count (Mono)  On-Time: 32 bytes  ex: 128-64-128-64-128-64-32-16-128-64-32-16-8-4-2-1-128-64-128-64-128-64-32-16-128-64-32-16-8-4-2-1  Data Format: 32 bytes  ex: 7-6-7-6-7-6-5-4-7-6-5-4-3-2-1-0- 7-6-7-6-7-6-5-4-7-6-5-4-3-2-1-0  Parameter는 다음그림과 같은 구조를 가진다. (data format에 따라 On-Time이 결정된다) |
| **Response** | <ESC>%STT<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000 |

**<< ON-TIME >>**

**Overlay, Mono 구조 (Overlay 8+3, Mono 8+3)**

**8 +3: 8bytes = On-Time (bit별 On-Time : 8bits: 1dot data를 8bits로 변경)**

**3bytes = Bit counts (**page 11 참조)

$SOP Command – Start Of Printing

|  |  |
| --- | --- |
| **Description** | 인쇄 시작을 설정한다.  이 명령어 뒤에 오는 명령어는 새로운 페이지의 데이터를 의미한다. |
| **Syntax** | <ESC>$SOP<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1 = Option. 0x00(Normal) |
| **Response** | <ESC>%SOP<status><length><CR>  <status:2>: Error/ACK (Error Code표 참조)  <length:4>: 00000000  : Error 46 추가 – Print Job pool Full (정상:0, FULL: 46) |

$SPT command – Set the print type

|  |  |
| --- | --- |
| **Description** | 인쇄 시 사용할 리본 타입, 인쇄 매수 등을 설정한다. |
| **Syntax** | <ESC>$SPT<length><data><CR> |
| **Parameter** | <length:4> **00000004**  <data:1> p1=리본 타입 설정  0x00: YMCKO (Full panel ribbon), 0x01: KO, 0x02:YMCKOK, 0x04:K ribbon  0x80:YMCKO (Half panel ribbon) , 0x06: Gold Ribbon, 0x0b: White Ribbon,  0x0c: Silver Ribbon, 0x0e: Red Ribbon, 0x0f: Blue Ribbon,  <data:2> p2=인쇄 매수 설정 (1~100)  이 값은 $PCP의 p1=0x00과 0x80 (All Color)일 때 동작한다.  **<data:1> p3=라미네이터 연결 동작설정**  **0x00: Laminator동작 수행을 하지 않는다. 0x01: Laminator동작 수행을 사용한다.** |
| **Response** | <ESC>%SPT<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

$CCB command – Clear Color/Varnish/Mono Image Buffer

|  |  |
| --- | --- |
| **Description** | 컬러/오버레이/모노 이미지 버퍼를 초기화한다. |
| **Syntax** | <ESC>$CCB<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data:1>p1=Buffer ID. 여러 개의 이미지 버퍼를 동시에 설정 가능하다.  Back-side buffer는 상위 bit를 사용한다(0x80)  0x01: Yellow Image Buffer  0x02: Magenta Image Buffer  0x04: Cyan Image Buffer  0x08: Dye Black Image Buffer  0x10: Varnish Image Buffer  0x20: Mono Image Buffer  0xFF: All Image Buffer (front, back 모두 clear)  0x3F: Front Image Buffer Clear (Front YMCKOK)  0xBF: Back-side Image Buffer Clear (Back YMCKOK)  <data:1>p2=버퍼 초기화 시 채울 값 (0x00 ~ 0xFF)  Overlay를 인쇄하지 않는 경우도 있으므로, Overlay도 전송되어야 한다. |
| **Response** | <ESC>%CCB<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

$TCB command – Color/Varnish/Mono Image Data Transfer (with parameters)

|  |  |
| --- | --- |
| **Description** | 지정한 이미지 버퍼(C, M, Y, K, O, M)에 인쇄 데이터를 다운로드한다. |
| **Syntax** | <ESC>$TCB<length><data><CR> |
| **Parameter** | <length:4> 0000XXXX (8+N)  <data:1>p1= Color ID  0x00=Image (JPEG Image: front-side jpg image 🡪 검토 후 가능 확인), 현재 미사용  0x01=Yellow, 0x02=Magenta, 0x04=Cyan, 0x08= Black(K)  0x10=Varnish  0x20=Mono  Back-side buffer는 상위 bit를 사용한다(0x80)  0x80 = Image (JPEG Image: back-side jpg image 🡪 검토 후 가능 확인), 현재 미사용  0x81, 0x82, 0x84, 0x88, 0x90, 0xA0 : Back-side image (YMCKOK)  <data:1>p2= Data mode  0x00 = Un-compressed Mode, 0x01 = JBIG (TBD), 0x02 = JPG (TBD)  <data:2>p3= Set resolution mode/gray mode  상위 1byte: TBD / 하위 1byte: data format  0x0000 = **default format** (1차 적용 format)  **- Color 32bits no-rotate / Mono 8bits no-rotate format**  0x0001 = **Original RAW format** (2차 적용 format)  **- Color 8bits no-rotate / Mono 1bits no-rotate format**  0x0002 = **Speed\_Image\_Optimized RAW format** (3차 적용 format)  **- Color 32bits Rotated / Mono 8bits Rotated format**  **p2가 Un-compress mode일 경우만 적용함.**  **p2가 JBIG/JPG 이면, p3는 don’t care (어떤 값이든 관계 없음)**  **전송되는 panel별로 p3가 다를 수 없으며, front/back모두 같은 값으로 설정됨**  <data:2>p4= Horizontal(X-axis) Start Position in dots (300dpi start) : 0x0010  <data:2>p5= Horizontal(X-axis) Width of graphic in dots (300dpi width) : 0x03EB  Mono/Varnish는 32bytes의 배수로 설정한다. (256, 512, 768,1024 dots)  <data:N>Compressed/Un-compressed Image data length  : data count는 <length> field에서 8을 뺀 만큼의 data이다. |
| **Response** | <ESC>%TCB<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

$PCP command – Print Color/Varnish/Mono Panel (a Panel / All Panel)

|  |  |
| --- | --- |
| **Description** | 선택 된 Color/Varnish/Mono Panel의 Image buffer를 Print한다. |
| **Syntax** | <ESC>$PCP<length><data><CR> |
| **Parameter** | <length:4> 00000003  <data:1> p1=color ID (panel)  0x00=front all color (ribbon 순서에 따라 color별 인쇄 함 all panel)  0x80=back all color (ribbon 순서에 따라 color별 인쇄 함 all panel)  0x01(0x81)=Yellow, 0x02(0x82)=Magenta, 0x04(0x84)=Cyan,  0x08(0x88)= Dye Sublimation Black(K)  0x10(0x90)=Varnish, 0x20(0xA0)=Mono  <data:1> p2=지정한 패널 인쇄 후 동작 모드 설정  0x00: 카드를 배출한다. (앞쪽)  **0x01: 한 판넬 (해당 판넬) SKIP**  **0x10: 카드를 인쇄 시작 위치로 리턴한다. (K, Color, …)**  **0x20: 카드를 그대로 둔다.**  **0x30: 카드를 뒤집는다. (changer로 card를 보내 뒤집고, 대기 상태로)**  **0x40: 카드를 체인저뒤로 보낸다. (뒤쪽 배출, 라미네이터로 보냄)**  **0x60: 후면으로만 배출하면서 카드가 물려 있는 상태로 멈춤(For Dispenser)**  **0x70: 후면으로만 배출하면서 카드가 물려 있지 않는 상태로 나감(For Dispenser)**  **0x80: 전면으로 배출하면서 카드가 물려있지 않는 상태로 나감(For Dispenser)**   * **상위 Nibble은 인쇄 후 동작, 하위 Nibble은 인쇄 전 동작을 나타냄.**   **해당 동작을 OR하여 사용 가능함.**  <data:1> p3= 인쇄 속도 설정  0x00=Fast mode  0x01=Normal mode  <data:1> p4 = 인쇄 Data 전송 모드 설정  0x00 = 1Pass Mode  0x01 = 3Pass Mode |
| **Response** | <ESC>%PCP<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000  🡺 Response 후, Host에서는 Status를 Check하여 동작 종료를 확인 할 수 있다. |

$POL command – Print Only Laminator (No printing)

|  |  |
| --- | --- |
| **Description** | Card를 Laminator 시작 위치로 보내고, 라미네이팅을 시작한다. (CIT 🡪 CHANGER 🡪Laminator)  Laminator에서 카드를 빨아들이고, Laminating을 시작하면 return한다. |
| **Syntax** | <ESC>$POL<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1> p1=option (reserved) |
| **Response** | <ESC>%POL<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000  🡺 Response 후, Laminator status를 확인하는 명령을 통해, 종료를 확인할 수 있다..  ACK: Laminator의 시작위치에서 시작함.  NAK: Laminator 시작 위치까지 가지 못하거나, Laminator에서 시작하지 못함. |
|  |  |

$CDT command – Calibrate Data Transfer

|  |  |
| --- | --- |
| **Description** | 선택 된 Color의 calibration data를 CIT로 전송한다. |
| **Syntax** | <ESC>$CDT<length><data><CR> |
| **Parameter** | <length:4> 0000XXXX (= 1 + N)  <data:1> p1= color ID (panel) (- $PCP, $TCB의 color ID와 동일)  Calibrated data는 Y, M, C, Kd, K(Mono)만 사용한다.  Overlay는 적용하지 않는다. Calibrated data 전송은 color별로 각각 수행한다.  0x01(0x81)=Yellow, 0x02(0x82)=Magenta, 0x04(0x84)=Cyan,  0x08(0x88)= Dye Sublimation Black(K)  0x20(0xA0)=Mono  <data: N> Calibrated data (Line 1~1024): 1 line calibrated data는 3074 level이다.  1line의 calibration data는 3bytes의 값으로 표현된다 (color\_rate1/color\_rate2/color\_level)  1color의 calibration data는 (MAX 3074bytes = 1024line x 3bytes + 2byte)이다.  Panel image 전송 후 ($TCB 명령이 전송된 후) 전송된다.  🡺 Ex: Y-M-.. : ($TCB-$CDT : Y-image) - ($TCB-$CDT: M-image) -…  핀트 알고리즘은 이미지 처리 하여 적용되므로 data를 전송할 필요가 없으며, 전송되는 data는 단차 알고리즘 가중치(color\_rate1/color\_rate2/color\_level) 이다. |
| **Response** | <ESC>%CDT<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000  🡺 ACK / NAK. |

$EOP Command – End Of Print

|  |  |
| --- | --- |
| **Description** | 인쇄 종료를 설정한다. – 한 Job (card에 인쇄되는 양면 Image)에 대한 종료 |
| **Syntax** | <ESC>$EOP<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=$SOP와 $EOP 명령 사이에 전송된 $PCP 명령 개수 (MAX 12개) |
| **Response** | <ESC>%EOP<status><length><data><CR>  <status:2> Error/ACK (Error code표 참조)  <length:4> 00000002  <data: 1> p1 = CIT에서 받은 $PCP count 개수 (정상: 명령의 p1와, 응답의 p1이 같음)  <data: 1> p2 = CIT에서 할당된 JOB-NO (cancel시 사용할 Job-No) |

$CPJ Command – Cancel Print Job

|  |  |
| --- | --- |
| **Description** | “$EOP” 이후, 인쇄가 종료되기 전에 인쇄 JOB를 cancel한다. |
| **Syntax** | <ESC>$CPJ<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1= CIT에서 할당된 JOB-NO. $EOP에서 회신된 JOB-NO |
| **Response** | <ESC>%CPJ<status><length><data><CR>  <status:2> Error/ACK (Error code표 참조)  <length:4> 00000000 |

$PJS Command – Print Job Status

|  |  |
| --- | --- |
| **Description** | 인쇄 상태를 확인하는 명령. |
| **Syntax** | <ESC>$PJS<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>%PJS<status><length><data><CR>  <status:2> Error/ACK (Error code표 참조)  <length:4> 00000007  <data: 1> p1 = 프린터 상태: Idle (0x00), 인쇄 중(0x01), 장애 발생-인쇄 중단 (No working=0xFF)  <data: 1> p2 = 인쇄 중인 JOB-NO. (장애시는 중단 상태의 Job-No)  <data: 1> p3 = 대기중인 JOB count (현재 동작중인 JOB 개수까지 포함)  - 장애발생시 인쇄중인 JOB도 대기중인 JOB으로 분류된다.  <data: 1> p4 = 현재 인쇄 중인 JOB의 copies (전체 인쇄 매수)  <data: 1> p5 = 남아있는 인쇄 매수 (remain count)  <data: 1> p6 = 재인쇄 확인창 표시 상태  - 0: 재인쇄 확인 창 사라짐  - 1: 재인쇄 확인 창(Auto 옵션) 표시  - 2: 재인쇄 확인 창 (Manual 옵션) 표시  <data: 1> p7 = 재인쇄 확인 창에 표시할 메시지 상태  - 1: 뚜껑을 열고 리본을 확인하세요  - 2: 카드 배출 중…  - 3: 리본 싱크 설정 중…  - 4: 카드를 제거해 주세요  - 5: 재인쇄하려면 확인 버튼을 누르세요. |
|  | <data: 1>p8 = Data 전송 가능 상태를 표시 (3pass 모드에서만 사용)   * 0x00 : Data 전송 불가 * 0x01 : Data 전송 가능 |

🡺 장애 발생(인쇄 중단) 시 JOB Cancel / Card Eject등을 수행한다. 장애 발생 후 다시 인쇄 Job을 Start하는조건을 장애가 clear되었을 때로 한다. (인쇄 중단 후 CARD EJECT + RIBBON SYNC 동작 후 인쇄 재 Start)

명령($RPE)을 받아서 재 시작할 수 한다. (카드 Eject나, 리본 SYNC를 하지 않아도 될 경우)

🡺 자동 장애 처리 옵션을 세팅( !SCV 명령에서 38번 항목 [0:Auot 옵션, 1: Manual 옵션]을 세팅)하면 다음의 방법으로 장애 처리를 한다.

- Auto 옵션: 에러 발생 후 표시되는 재인쇄 확인 창에 장애 처리 및 상태 메시지를 표시하며 자동 장애 처리 완료 후 재인쇄 여부($RPE or $CPJ)를 확인하고 처리한다.

- Manual 옵션: 에러 발생 후 표시되는 재인쇄 확인 창에서 재인쇄 여부 ($RPE or $CPJ)를 확인하고 처리한다.

      (재인쇄 확인 창 표시상태가 0이 오거나 $RPE나 $CPJ에 대한 응답이 오면 재인쇄 확인 창이 사라진다.)

$RPE Command – Re-Print start after Error

|  |  |
| --- | --- |
| **Description** | 인쇄중단 후 강제로 인쇄를 재 시작할 때 인쇄 START 명령. |
| **Syntax** | <ESC>$RPE<length><CR> |
| **Parameter** | <length:4> 00000000 |
| **Response** | <ESC>%RPE<status><length><CR>  <status:2> Error/ACK (Error code표 참조)  <length:4> 00000000 |

🡺 장애 발생(인쇄 중단) 후 인쇄 재 시작 시 중단되었던 JOB부터 다시 시작한다. 단 인쇄 매수 관련하여서는 인쇄매수에서 인쇄 완료된 매수는 제외하고, 나머지만 인쇄 한다.

4.5 SetUp Command

+LDD Command – LCD Display Data (16 character/line)

|  |  |
| --- | --- |
| **Description** | 현재 저장 되어 있는 Varnish buffer의 Graphic을 프린트 한다 |
| **Syntax** | <ESC>+LDD<length><data><CR> |
| **Parameter** | <length:4> 0000XXXX (2+N)  <data: 1> p1 = LCD Line Number: 0x01, 0x02, 0x03, 0x04 – 각 line number  <data: 1> p2 = LCD Line Reverse Option  0x00= p1으로 설정 된 LCD Line Normal Display  0x01 = p1으로 설정 된 LCD Line Reverse Display  <data: N> display data  영문: ASCII Code 중 Control Character 제외.  한글: 조합형 한글 Code. |
| **Response** | <ESC>=LDD<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

+LDO Command – LCD Display Option command

|  |  |
| --- | --- |
| **Description** | 4 Line 16 Char LCD의 Display Option |
| **Syntax** | <ESC>+LDO<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data: 1> p1 = LCD Line Option  0x30 ⇒ ALL LINE Clear  0x31 ⇒ LINE 1 Clear  0x32 ⇒ LINE 2 Clear  0x33 ⇒ LINE 3 Clear  0x34 ⇒ LINE 4 Clear  0x35 ⇒ TP9000 Default LCD Display  0x38 ⇒ LCD Backlight ON  0x39 ⇒ LCD Backlight OFF |
| **Response** | <ESC>=LDO<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

+LUR Command – LCD User memory Read (Read LCD memory)

|  |  |
| --- | --- |
| **Description** | Flash의 사용자용 LCD 메모리에서 LCD 라인 별 정보 및 Data를 읽는다. |
| **Syntax** | <ESC>+LUR<length><data><CR> |
| **Parameter** | <length:4> 00000002  <data:1> p1=메모리 타입  0x00: RAM(기본메모리) – 현재 사용 안함  0x11: Flash(기본메모리)  <data:1> p2=LCD Line Number (0x31 ~ 0x34 : Line 1, 2, 3, 4) |
| **Response** | <ESC>=LUR<status><length><data><CR>  <status:2> Error/ACK  <length:4> 00000XXX (1+N)  <data:1>p1= data format  : 0x30 – Text, 0x40 - Text(화면 반전) , 0x31 - Image, 0x41 - 미 지정  <data: N> LCD data  Text : 16 bytes(ASCII code), Image : 256 bytes  Ex) <ESC>=LUR<status><00000011><0x30, TITENG1234567890><CR> : Text |

+LUW Command – LCD User memory Write (Write data into LCD memory)

|  |  |
| --- | --- |
| **Description** | Flash의 사용자용 LCD 메모리에 LCD 라인 별 정보 및 Data를 쓴다.. |
| **Syntax** | <ESC>+LUW<length><data><CR> |
| **Parameter** | <length:4> 00000XXX (3+N)  <data:1>p1=메모리 타입  0x00: RAM(기본메모리) – 현재 사용 안 함  0x11: Flash(기본메모리)  <data:1>p2=LCD Line Number (0x31 ~ 0x34 : Line 1, 2, 3, 4)  <data:1>p3=Data Format  : 0x30 = Text, 0x40 = Text(화면 반전), 0x31 = Image, 0x41 = Erase (p1’s line)  <data: N> Data |
| **Response** | <ESC>=LUW<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

[Appendix] 사용자 LCD 읽기/쓰기에 사용할 Text 또는 Image 파일 구조

\* LCD 정보는 Flash 메모리에 저장 가능합니다.

\* LCD 크기: 128 x 64 pixels로 구성됨.

\* 한 라인 크기: 텍스트 = 16 x 1 = 16바이트, 이미지 = 16(128/8) x 16 = 256바이트

**<< Text 파일 구조 >>**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Format | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | D16 | CR | LF |
| Format | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | D16 | CR | LF |
| Format | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | D16 | CR | LF |
| Format | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | D16 | CR | LF |

\* Data Format 설정: Text일 때 0x30(‘0’)이고, Text(반전)일 때는 0x40(‘@’)이다.

\* CR=0x0D, LF=0x0A, SP=0x20(space)

\* 모든 라인은 CR과 LF로 끝나야 한다.

\* User LCD Memory Write일 때, LCD Line에 Data가 없으면 CR, LF만 보낸다.

예: Line1=”CARD PRINTER”(반전), Line2=”TP-9000”, Line3=데이터 없음, Line3=”2009.03.16(V116)”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| @ | C | A | R | D |  | P | R | I | N | T | E | R |  |  |  |  | CR | LF |
| 0 | T | P | - | 9 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  | CR | LF |
| SP | CR | LF |
| 0 | 2 | 0 | 0 | 9 | . | 0 | 3 | . | 1 | 6 | ( | V | 1 | 1 | 6 | ) | CR | LF |

\* User LCD Memory Read일 때, LCD Line에 Data가 없으면 SP로 초기화하여 16 바이트를 채운다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| @ | C | A | R | D |  | P | R | I | N | T | E | R | SP | SP | SP | SP | CR | LF |
| 0 | T | P | - | 9 | 0 | 0 | 0 | SP | SP | SP | SP | SP | SP | SP | SP | SP | CR | LF |
| SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | CR | LF |
| 0 | 2 | 0 | 0 | 9 | . | 0 | 3 | . | 1 | 6 | ( | V | 1 | 1 | 6 | ) | CR | LF |

**<< Image 파일 구조 >>**

\* 일반 Bitmap 저장 구조와 동일함.

\* User LCD Memory Read일 때, LCD Line에 Data가 없으면 0xFF(white)로 초기화하여 256 바이트를 채운다.

|  |
| --- |
| Bitmap Header |
| Image Data (LCD 4 line) – 256 bytes |
| Image Data (LCD 3 line) – 256 bytes |
| Image Data (LCD 2 line) – 256 bytes |
| Image Data (LCD 1 line) – 256 bytes |

+UMC Command – Clear User’s Memory

|  |  |
| --- | --- |
| **Description** | Clear User’s Memory |
| **Syntax** | <ESC>+UMC<length><data><CR> |
| **Parameter** | <length:4> 00000001  <data:1>p1=사용자 메모리 타입  0x00: RAM(기본메모리) – 128바이트 지원. 메모리를 0xFF로 초기화  0x01: RAM(기본메모리) – 500KB 지원. 메모리를 0xFF로 초기화  0x11: Flash(기본메모리) – 4KB 지원. 메모리를 0xFF로 초기화 |
| **Response** | <ESC>=UMC<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

+UMR Command – Get(Read) User’s Memory

|  |  |
| --- | --- |
| **Description** | 사용자 메모리 영역에서 지정된 offset부터 지정된 길이만큼 data를 읽어온다 |
| **Syntax** | <ESC>+UMR<length><data><CR> |
| **Parameter** | <length:4> 00000009  <data:1>p1=사용자 메모리 타입  0x00: RAM(기본메모리) – 128바이트 지원 (0x00000080)  0x01: RAM(기본메모리) – 500KB 지원 (0x0007D000)  0x11: Flash(기본메모리) – 4KB 지원 (0x00001000)  <data:4>p2=offset (0x0000 ~ N-1)  <data:4>p3=length (0x0001 ~ N) |
| **Response** | <ESC>=UMR<status><length><data><CR>  <status:2> : Error/ACK (Status of CIT)  <length:4>: 00XXXXXX (1+N)  <data:1>p1=사용자 메모리 타입  <data:N>User Memory의 해당 위치(offset)의 데이터 (command의 length만큼 읽어옴) |

+UMW Command – Write User’s Memory

|  |  |
| --- | --- |
| **Description** | 사용자 메모리 영역의 지정된 위치(offset)로부터 지정된 길이(length)만큼 data를 쓴다 |
| **Syntax** | <ESC>+UMW<length><data><CR> |
| **Parameter** | <length:4> 0000XXXX (9+N)  <data:1>p1=사용자 메모리 타입  0x00: RAM(기본메모리) – 128바이트 지원 (0x00000080)  0x01: RAM(기본메모리) – 500KB 지원 (0x0007D000)  0x11: Flash(기본메모리) – 4KB 지원 (0x00001000)  <data:4>p2=offset (0x0000 ~ N-1)  <data:4>p3=length (0x0001 ~ N)  <data:N>User memory에 write해야 할 data |
| **Response** | <ESC>=UMW<status><length><CR>  <status:2> Error/ACK  <length:4> 00000000 |

# 5. **Status / Error Code**

## **5.1 Status Code**

Status Code: Response Frame의 “Status” field의 값으로, 상태 및 error code값.

|  |  |  |
| --- | --- | --- |
| **분류** | **Status Value** | **설명** |
| ACK (OK) | 00 00 | ACK – 정상 동작 중 – 명령어 수신 후 동작 중. (대부분 미 사용)  Mecha 동작 or User동작 중 |
|  | 00 06 | ACK – 정상 동작 완료. |
| NACK | 00 15 | NACK – NACK |
| Information | 10 XX | **추가 보충 – 진행하면서 추가 함. (TBD)**  **0x1001 – Printing 중**  **0x1002 - MS동작 중**  **0x1004 – IC 동작 중** |
| Error/Warning | 8X XX | MS / IC / RF / SAM / Ribbon / TPH의 error 정보 |
| Error/Warning | 4X XX | DLL에서 설정하는 오류 코드 - USB 에러, DLL 처리 오류 |

## **5.2 Error Code**

Error Code역시, Response Frame의 “Status” field에 나타낸다. (FW Code: “enum CIT\_ALL\_ERRORS\_e”)

Error 발생시 Status field에 “80 XX”로 나타낸다. (80: Hexa value, XX: ASCII)

| **Code** | | **FW Define** | **장애 내용** | **설명** |
| --- | --- | --- | --- | --- |
| 80 | 10 | TPH\_OPEN\_ERROR | TPH Module Open Error | 프린트 시 TPH Module Open |
| 80 | 11 | HEAD\_ERROR | TPH Up/Down Error | TPH Up/Down 실패 |
| 80 | 12 | PATH\_ERROR | Path Error | Card Feeding시 Jam |
| 80 | 13 | TPH\_TEMP\_ERROR | TPH Temperature Error | TPH 써미스터 온도 비 정상 |
| 80 | 14 | RIBBON\_ERROR | Ribbon Encoder Error | 리본 Encoder 미입력 |
| 80 | 15 | MEDIA\_ERROR | Media Error | 비정상 Media 입력 |
| 80 | 16 | EEPROM\_ERROR | EEPROM Fail |  |
| 80 | 17 | IC\_UP\_DOWN\_ERROR | IC Module Up/Down Fail |  |
| 80 | 18 | HOPPER\_CAM\_ERROR | Hopper CAM Error |  |
| 80 | 20 | CMD\_ERROR | Command Error | Invalid Command |
| 80 | 21 | INVALID\_PARA | Invalid Parameter | Invalid Parameter |
| 80 | 22 | TIME\_OUT | Time Out | 주 통신부 Time Out |
| 80 | 23 | IC\_CONTACT\_ERROR | IC Contact Error | Ic Contact 불량 |
| 80 | 24 | ENCODE\_ERROR | MS Encode Error |  |
| 80 | 25 | MS\_READ\_ERROR | MS Read Error |  |
| 80 | 26 | STX\_ERROR | MS STX Error |  |
| 80 | 27 | PARITY\_ERROR | MS Parity Error |  |
| 80 | 28 | ETX\_ERROR | MS ETX Error |  |
| 80 | 29 | LRC\_ERROR | MS LRC Error |  |
| 80 | 30 | IC\_TIMEOUT | IC Module Time Out | IC Module 통신 Time Out |
| 80 | 31 | RF\_TIMEOUT | RF Module Time Out | RF Module 통신 Time Out |
| 80 | 32 | SAM\_TIMEOUT | SAM Module Time Out | SAM Module 통신 Time Out |
| 80 | 33 | CHANGER\_TIMEOUT |  |  |
| 80 | 34 | CARD\_INPUT\_TIMEOUT | Card Input Time Out | Hopper에서 카드 삽입 시 발생 한 Time Out |
| 80 | 35 | RIB\_NEAR\_END | Ribbon Near End | Ribbon Near End |
| 80 | 36 | RIB\_END | Ribbon End | Ribbon End |
| 80 | 37 | NO\_MS\_DATA | No MS Data |  |
| 80 | 38 | RIB\_ERROR | RF-Tag Ribbon Error | 인쇄 명령 수신 후 비 정상 리본 장착 시 인쇄 시작 전 장애 발생 |
| 80 | 39 | USER\_FLASH\_ERROR | User Memory Error | 사용자용 메모리 초기화/읽기/쓰기 장애 발생 |
| 80 | 40 | MS\_JITTER\_ERROR | MS Jitter Error |  |
| 80 | 41 | RFTAG\_TIMEOUT | RF TAG TIMEOUT | RF TAG 통신 Time Out |
| 80 | 42 | INCORRECT\_ERROR | Incorrect Ribbon | 인쇄 명령 수신 후 비 정상 리본 장착 시 인쇄 시작 전 장애 발생 |
| 80 | 43 | PRINTER\_LOCKED | Printer Locked |  |
| 80 | 44 | TDES\_ERROR | Triple-DES Error |  |
| 80 | 45 | PASSWORD\_ERROR | Password Error |  |
| 80 | 46 | PRINT\_JOB\_FULL | Print Job Pool Full | $SOP의 응답 값으로 올라올 수 있음 |
| 80 | 47 | LMIF\_TIMEOUT | Laminator I/F timeout error |  |
| 80 | 48 | LMIF\_NOTREADY | Laminator is not ready |  |
| 80 | 50 | CH\_TIMEOUT\_ERR | Changer Module Time Out | Changer Module 통신 Time Out |
| 80 | 51 | CH\_ACK\_TIMEOUT\_ERR | Changer Ack Time Out |  |
| 80 | 52 | CH\_HOME\_POS\_ERROR | Changer Home Position Fail |  |
| 80 | 53 | CH\_TURN\_UP\_ERROR | Changer Turn Up Fail |  |
| 80 | 54 | CH\_CARD\_INPUT\_TIMEOUT | Changer Card Input Time Out | Hopper에서 카드 인입시 시간 초과 |
| 80 | 55 | CH\_CARD\_INPUT\_JAM | Changer Card Input Jam | Changer로 카드 인입 진입시 잼 발생 |
| 80 | 56 | CH\_CARD\_JAM | Changer Card Output Jam | Changer에서 카드 배출시 잼 발생 |
| 80 | 57 | CH\_CHAR\_EXIST | Changer Card Inside | 메카 초기화 시 Changer가 비정상 위치, 카드 내부에 존재 시 |
| 80 | 60 | MS\_INVALID\_TRACK\_NO | Invalid track number |  |
| 80 | 61 | MS\_INVALID\_DATA\_TRACK1 | Invalid data track 1 |  |
| 80 | 62 | MS\_INVALID\_DATA\_TRACK2 | Invalid data track 2 |  |
| 80 | 63 | MS\_INVALID\_DATA\_TRACK3 | Invalid data track 3 |  |
| 80 | 64 | MS\_INVALID\_LENGTH\_TRACK1 | Invalid length track 1 |  |
| 80 | 65 | MS\_INVALID\_LENGTH\_TRACK2 | Invalid length track 2 |  |
| 80 | 66 | MS\_INVALID\_LENGTH\_TRACK3 | Invalid length track 3 |  |
| 80 | 67 | MS\_INVALID\_DATA\_JIS\_II | Invalid data JIS2 | JIS2 |
| 80 | 68 | MS\_INVALID\_LENGTH\_JIS\_II | Invalid length JIS2 | JIS2 |
| 80 | 70 | IMAGE\_INVALID | Invalid Image |  |
| 80 | 71 | IMAGE\_PROCESS\_ERROR | Image processing fail |  |
| 80 | 72 | TERMINAL\_NOT\_LOCKED | Terminal not locked | Terminal Lock 설정 실패 |
| 80 | 81 | RF\_CMD\_OPERATING\_ERROR | RF 모듈 오류: 0x01 |  |
| 80 | 82 | RF\_CMD\_LENGTH\_ERROR | RF 모듈 오류: 0x02 |  |
| 80 | 83 | RF\_CMD\_BCC\_ERROR | RF 모듈 오류: 0x03 |  |
| 80 | 84 | RF\_CMD\_TIMEOUT\_ERROR | RF 모듈 오류: 0x04 |  |
| 80 | 85 | RF\_ERROR\_85 | Not defined |  |
| 80 | 86 | RF\_ERROR\_86 | Not defined |  |
| 80 | 87 | RF\_ERROR\_87 | Not defined |  |
| 80 | 88 | RF\_CARD\_OVERLAP\_ERROR | RF 모듈 오류: 0x80 |  |
| 80 | 89 | RF\_CMD\_RESPONSE\_NOT\_VALUE | RF 모듈 오류: 0xFF |  |
| 80 | 90 | SCAN\_IDLE\_STATUS | (No error - status) |  |
| 80 | 91 | SCAN\_READY | (No error - status) |  |
| 80 | 92 | SCAN\_CARD\_TIMEOUT |  |  |
| 80 | 93 | SCAN\_CALIBRATION |  |  |
| 80 | 94 | SCAN\_STARTING | (No error - status) |  |
| 80 | 95 | SCAN\_SCANNING | (No error - status) |  |
| 80 | 96 | SCAN\_SCANNED | (No error - status) |  |
| 80 | 97 | SCAN\_ERROR |  |  |
| 80 | 98 | API\_NOT\_SUPPORT | (FW – not used) |  |
| 80 | 99 | UNKNOWN\_ERROR |  |  |
| 80 | 120 | [IC] Success | (No error - status) |  |
| 80 | 121 | IC\_UNKNOWN\_DRV\_OR\_CMD | 01h: Unknown Driver or Command |  |
| 80 | 122 | IC\_OPER\_IMPOS\_THIS\_DRV | 02h: Operation Impossible with This Driver |  |
| 80 | 123 | IC\_INCORRECT\_NO\_ARG | 03h: Incorrect Number Of Arguments |  |
| 80 | 124 | IC\_READER\_COMMAND\_UNKNOWN | 04h: Invalid Command Code |  |
| 80 | 125 | IC\_RES\_TOO\_LONG\_BUFFER | 05h: Response Too for the Buffer |  |
| 80 | 126 | IC\_RES\_ERR\_CARD\_RESET | 10h: Response Error at the Card Reset |  |
| 80 | 127 | IC\_MESSAGE\_TO\_LONG | 12h: Message Too Long. |  |
| 80 | 128 | IC\_BYTE\_READ\_ERROR | 13h: Byte reading Error returned by an asynchronous card |  |
| 80 | 129 | IC\_CARD\_POWERED\_DOWN | 15h: Card Powered down. |  |
| 80 | 130 | IC\_INCORRECT\_NO\_PARAM | 1Bh: Incorrect Number of Parameter. |  |
| 80 | 131 | IC\_TCK\_CHECK\_BYTE\_INCORRECT | 1Dh: The TCK check byte is incorrect in a microprocessor card Answer To Reset. |  |
| 80 | 132 | IC\_ERR\_CARD\_RESET\_RES | A0h: Error in the card reset response, such as unknown exchange protocol, or TA1 byte not recognized. |  |
| 80 | 133 | IC\_CARD\_PROTO\_ERROR | A1h: Card protocol error (T=0/T=1). |  |
| 80 | 134 | IC\_CARD\_MALFUNCTION | A2h: Card malfunction. The card does not respond to the reset or has interrupted an exchange by timing out. |  |
| 80 | 135 | IC\_PARITY\_ERR\_MP\_EXCHANGE | A3h: Parity error during a microprocessor exchange. |  |
| 80 | 136 | IC\_CARD\_ABORTED\_CHAINING | A4h: Card has aborted chaining (T=1). |  |
| 80 | 137 | IC\_READER\_ABORTED\_CHAINING | A5h: Reader has aborted chaining (T=1). |  |
| 80 | 138 | IC\_RESYNCH\_SUCCESS | A6h: RESYNCH successfully performed by GemCore. |  |
| 80 | 139 | IC\_PROTO\_PARAM\_SEL\_ERROR | A7h: Protocol Parameter Selection (PPS) error. |  |
| 80 | 140 | IC\_CARD\_POWERD\_UP | A8h: Card already powered up. |  |
| 80 | 141 | IC\_PCLINK\_CMD\_NOT\_SUPPORT | B0h: PC-Link command not supported. |  |
| 80 | 142 | IC\_PROCEDURE\_BYTE | E4h: The card has just sent an invalid "Procedure Byte" (see ISO 7816-3). |  |
| 80 | 143 | IC\_INTERRUPTED\_EXCHANGE | E5h: The card has interrupted an exchange (the card sends an SW1 byte but more data remains to be sent or received). |  |
| 80 | 144 | IC\_ERROR\_144 | - |  |
| 80 | 145 | IC\_ERROR\_RETURNED\_CARD | E7h: Error returned by the card. The SW1 and SW2 bytes returned by the card are different than 90h 00h. |  |
| 80 | 146 | IC\_CARD\_REMOVED | F7h: Card removed. The card has been withdrawn during the execution of a command. Check that the card instruction is not partially completed. |  |
| 80 | 147 | IC\_CONSUMING\_CIRCUITING | F8h: The card is consuming too much electricity or is short-circuiting. |  |
| 80 | 148 | IC\_CARD\_MISSING | FBh: Card missing. There is no card in the smart card interface. |  |
| 80 | 149 | IC\_WRONG\_CMD\_LENGTH | 01h: Wrong command length. (GemCore Pos Pro) |  |
| 80 | 150 | IC\_SLOT\_NO\_INVALID | 05h: The slot number is invalid (it must be set to 0 to 4). (GemCore Pos Pro) |  |
| 80 | 151 | IC\_BYTE\_DISPLAYED\_INVALID | 07h,08h,09h,15h,0Ah: The byte displayed is invalid. (GemCore Pos Pro) |  |
| 80 | 152 | IC\_ERROR\_152 | - |  |
| 80 | 153 | IC\_ATR\_TOO\_LONG | A3h: The ATR is too long (the number of bytes is greater than 33). (GemCore Pos Pro) |  |
| 80 | 154 | IC\_WRONG\_APDU\_LENGTH | BEh: The APDU command length is wrong. (GemCore Pos Pro) |  |
| 80 | 155 | IC\_TS\_OF\_ATR\_INVALID | F8h: The first byte (TS) of the ATR is invalid (reader in PC/SC - ISO mode). (GemCore Pos Pro) |  |
| 80 | 156 | IC\_RCV\_DATA\_FROM\_CARD | FCh: The reader receives a data from the card while it sends data to the card. (GemCore Pos Pro) |  |
| 80 | 157 | IC\_CARD\_NOT\_PRESENT\_MUTE | FEh: The card is not present or it is mute. (GemCore Pos Pro) |  |
| 80 | 158 | IC\_ERROR\_158 | - |  |
| 80 | 159 | IC\_ERROR\_158 | ??h: Not defined |  |
| 80 | 160 | IC\_MEM\_INVALID\_HANDLE | [ICMEM] Incorrect smart card handle |  |
| 80 | 161 | IC\_MEM\_INVALID\_PARAMETERS | [ICMEM] Incorrect Parameters |  |
| 80 | 162 | IC\_MEM\_INVALID\_LENGTH | [ICMEM] Incorrect Length |  |
| 80 | 163 | IC\_MEM\_INVALID\_ADDRESS | [ICMEM] Incorrect Address |  |
| 80 | 164 | IC\_MEM\_EARLY\_END\_OF\_FILE | [ICMEM] Incorrect length vs address |  |
| 80 | 165 | IC\_MEM\_PROTOCOL\_NOT\_SUPPORTED | [ICMEM] Protocol not supported |  |
| 80 | 166 | IC\_MEM\_NO\_READERS\_AVAILABLE | [ICMEM] No Reader Available |  |
| 80 | 167 | IC\_MEM\_UNKNOWN\_CARD | [ICMEM] Unknown Card (No ATR) |  |
| 80 | 168 | IC\_MEM\_UNKNOWN\_ERROR | [ICMEM] Unknown Error |  |
| 80 | 169 | IC\_MEM\_CARD\_UNSUPPORTED | [ICMEM] Incorrect (sub) card type |  |
| 80 | 170 | IC\_MEM\_INSUFFICIENT\_BUFFER | [ICMEM] Buffer length too small |  |
| 80 | 171 | IC\_MEM\_NO\_SMARTCARD | [ICMEM] No ICC in the contact unit. |  |
| 80 | 172 | IC\_MEM\_NO\_ACCESS | [ICMEM] Access not possible |  |
| 80 | 173 | IC\_MEM\_WRITE\_ERROR | [ICMEM] Writing unsuccessful |  |
| 80 | 174 | IC\_MEM\_WRONG\_PWD | [ICMEM] Wrong password |  |
| 80 | 175 | IC\_MEM\_NO\_SERVICE | [ICMEM] Resource Manager unavailable |  |
| 80 | 176 | IC\_MEM\_INVALID\_FIRMWARE\_ANSWER | [ICMEM] Invalid Firmware Answer |  |

## **5.3 DLL Error Code**

DLL에서 발생하는 오류 코드를 정의한다. 장치 오류 코드(USB, Network, …)와 DLL 처리 오류 코드

Error 발생시 Status field에 “40 XX”로 나타낸다. (40: Hex, XX: ASCII)

| **Code** | | **장애 내용** | **설명** |
| --- | --- | --- | --- |
| 40 | 4 | Response timeout | 성공 |
| 40 | 5 | Response Data Frame Error |  |
| 40 | 1001 | Device Invalid Handle |  |
| 40 | 1002 | Device Not Found |  |
| 40 | 1003 | Device Not Opened |  |
| 40 | 1004 | Device I/O Error |  |
| 40 | 1017 | Device API Not Found | 장치 DLL이 지원하는 않는 함수를 호출했을 때 |
| 40 | 1019 | Device List Not Ready |  |
| 40 | 1040 | Device DLL Not Found | (ex: FTD2XX.DLL) |
| 40 | 10xx | Device Error Occurred | 정의되지 않은 장치 오류 발생했을 때 |