CRT-350 V3.0 Card Reader/Writer Communication Protocol

Contents

1. Basic Transmission Specification ................................................ 5
2. Transmission Control Method ........................................................ 6
3. Transmission Format and Characters .............................................. 7
4. Transmission Control Procedure ................................................... 8
   4.1 Normal Operation(command and response) ....................................... 8
   4.2 Sequence in abnormal operation (command and response) ............ 8
5. Command/Response ........................................................................ 10
   5.1 Command Format (HOST->UNIT) ................................................... 10
   5.2 Positive response format(ICRW->HOST) ....................................... 10
   5.3 Negative response format(ICRW->HOST) ...................................... 10
6. CRT-350 Executed code area(status code and error code) .................. 12
   6.1 Command list .............................................................................. 12
   6.2 Status Code ................................................................................. 17
   6.3 Error Code .................................................................................. 17
7. Structure and Location Schematic Of CRT-350 ................................... 20
8. Detailed Description of CRT-350 Commands ...................................... 21
   8.1 Initialize command ........................................................................ 21
   8.2 Status request command ............................................................... 24
      8.2.1 Require status ........................................................................ 24
      8.2.2 Require intake status of ICRW ............................................... 27
   8.3 Entry command ............................................................................ 29
      8.3.1 Entry for once ....................................................................... 29
8.3.2 Entry for many times ................................. 30
8.3.4 Special command of jittering card in/out ............................................. 32
8.4 Carry card command............................................ 33
  8.4.1 Carry card .................................................. 33
  8.4.2 Expansion Carry card .................................... 34
8.5 Retrieve command .................................................. 35
8.6 MAG-Track Read operation command ......................... 36
  8.6.1 MAG-Track Read command ............................ 36
  8.6.2 MAG-Track special read command .................... 39
  8.6.3 Multi-MAG-Track read command .................... 40
8.7 MAG-Track write operation command ....................... 43
  8.7.1 MAG-Track write command (1) ..................... 43
  8.7.2 MAG-Track write command (2) ..................... 45
  8.7.3 MAG-Track write command (3) ..................... 46
  8.7.4 MAG-Track write command (4) ..................... 48
  8.7.5 MAG-Track write command (5) ..................... 49
8.8 Photoelectric sensor detect (No support for CRT-350) ....... 50
8.9 IC Contact command ............................................. 51
8.10 Revision read command ....................................... 52
8.11 Retract counter command .................................... 53
  8.11.1 Retract counter read .................................. 53
8.12 CPU card control command .................................... 55
  8.12.1 Activate CPU card command (cold reset) .......... 55
  8.12.2 Deactivate CPU card command ..................... 57
  8.12.4 C-APDU (T=0) ........................................ 59
  8.12.6 CPU send link block data as T=1(“C15”) ........ 62
  8.12.7 CPU send the last link block data as T=1(“C16”) .... 63
  8.12.8 CPU receive the link block data as T=1(“C17”) .... 64
  8.12.9 CPU warm reset ........................................ 65
  8.12.10 CPU card automatic communication (C-APDU command) ........ 66
8.13 SAM (Secure Application Module) control command .......... 67
  8.13.1 Activate SAM command (cold reset) .............. 67
  8.13.2 Deactivate SAM command ............................. 69
  8.13.3 Inquire SAM Status and working frequency command ......... 70
  8.13.4 SAM communication T=0 ............................... 72
  8.13.5 SAM communication T=1 ............................... 73
  8.13.6 SAM send link block data as T=1(“CIE”) ........ 75
  8.13.7 SAM send the last link block data as T=1(“CIF”) .... 76
  8.13.8 SAM receive the link block data as T=1(“CIG”) .... 77
  8.13.9 SAM warm reset ........................................ 78
  8.13.10 SAM card automatic communication (C-APDU command) .......... 79
8.13.11 Select SAM................................................................. 80
8.14 Monitoring for removal command........................................... 81
8.15 On/Off for red and green lights control command...................... 82
8.16 Anti-fishing control (just for E shutter)........................................ 83
  8.16.1 Anti-fishing status .......................................................... 83
  8.16.2 Anti-fishing mode setting.................................................... 83
  8.16.3 Anti-fishing pin control command......................................... 84
  8.16.4 Anti-fishing unit setting..................................................... 85
8.17 Automatically Detect Card Type ............................................ 87
  8.17.1 Automatically detect IC card type......................................... 87
  8.17.2 Automatically detect RF card type........................................ 89
8.18 EEPROM read/write(omit)................................................... 90
8.19 Lifetime of entry, MAG. head, IC card device............................ 91
  8.19.1 Gain the Lifetime of entry, MAG. head, IC card device.............. 91
  8.19.2 Initialize the Lifetime of entry, MAG. head, IC card device........ 92
8.20 Read/write ICRW's SN.......................................................... 93
  8.20.1 Read ICRW SN.............................................................. 93
  8.20.2 Write ICRW's SN(omit).................................................. 94
8.21 Read Configured characters.................................................. 95

Annex I Contact IC card control command................................................ 96
Annex II Siemens memory card control command.............................. 108
Annex III Contactless MIFARE ONE card control command................. 124
Annex IV Explanation of error code.................................................. 1399
Annex V Values of ATR parameter (TA1 and TA2).............................. 1463
Annex VI C-APDU Format.............................................................. 1499
Annex VII Calculation method of CRCC............................................ 1500
Annex VIII Magnetic tracks data.................................................... 1522

Note) The specification of CRT-350 V2.1 and V3.0 is a new version based on the compatible with SANKYOICT3Q8-3A0XXX, also, the blue words denotes the different or additional parts in CRT against SANKYO.
## Specification History of Modification

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2008.8.2</td>
<td>Draft</td>
</tr>
<tr>
<td>1.1</td>
<td>2008.10.15</td>
<td>Addition of the option about CPU card protocol(ISO7816/EMV2000) and cleanup of counter, and Modification of errors in some parts</td>
</tr>
<tr>
<td>2.0</td>
<td>2009.2.11</td>
<td>The bran-new version of compatible with SANKYO</td>
</tr>
</tbody>
</table>
| 2.1  | 2009.9.11  | 1. Firmware amend for WOSA operation;  
2. Amend on parity bit, see “1.”;  
3. Add “Jitter” command list, see‘8.3.3’;  
4. Add anti-fishing function command, see ‘8.16’;  
5. Amend on‘8.20.1’ |
| 3.0  | 2010.1.19  | 1. Add and amend ‘8.17’ about automatically check card type of TYPE A/B CPU, change command parameter from ‘31H’ to ‘32H’.  
2. Add and amend Annex III, changing command parameter from ‘92H’ to ‘93H’ and add TYPE A/B CPU card command. |
# 1 Basic Transmission Specification

<table>
<thead>
<tr>
<th>Item</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous Method</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>Communication Method</td>
<td>Half-duplex</td>
</tr>
<tr>
<td>Transmission Speed</td>
<td>9,600BPS, 19,200BPS, 38,400BPS, or 57,600BPS (automatic recognition)</td>
</tr>
<tr>
<td>Character Format*1</td>
<td>Start Bit 1bit</td>
</tr>
<tr>
<td></td>
<td>Data 8bit</td>
</tr>
<tr>
<td></td>
<td>Parity Bit 1bit (odd)</td>
</tr>
<tr>
<td></td>
<td>Stop Bit 1bit</td>
</tr>
<tr>
<td>Character Encode</td>
<td>ASCII 8bit code</td>
</tr>
</tbody>
</table>

*1 Format Figure

<table>
<thead>
<tr>
<th>Start Bit</th>
<th>D0</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
<th>P</th>
<th>Stop Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2 Transmission Control Method

- Command/Response Method
- The UNIT executes processes in accordance with commands sent by the host
- Results obtained from those processes are sent to the host as responses to those commands
3 Transmission Format and Characters

<table>
<thead>
<tr>
<th>STX (0xF2)</th>
<th>LEN (2byte)</th>
<th>TEXT</th>
<th>CRCC (2byte)</th>
</tr>
</thead>
</table>

STX (F2H) Indicate start of text STX code is F2H.
ACK (06H) Acknowledge.
NAK (15H) Negative acknowledge.
DLE, EOT (10H, 04H) Clear the line.
LEN(2bytes) Text length.
TEXT Command or response.
CRCC(2bytes) Cyclic redundancy code. (Polynomial X16+ X12+X5+1. Initial value is 0000H)
4. Transmission Control Procedure

4.1 Normal Operation (command and response)

4.2 Sequence in abnormal operation (command and response)

Case 1

Case 2

Case 3
ICRW  

Case 4

Host  

ICRW  

Command  

20msec Timeout (except entry, initialize command)

Command  

Execution  

ACK  

Response  

Response  

ACK  

Case 5

ICRW  

Command  

DLE, EOT  

DLE, EOT  

Host  

ACK  

Execution  

discontinue  

DLE, EOT  

Case 6

(HOST)  

Command  

DLE, EOT  

DLE, EOT  

300msec Timeout  

X error  

Discontinue  

ICRW  

ACK  

(Execution)  

Case 7

(HOST)  

Command  

DLE, EOT  

X error  

ACK  

ICRW  

ACK  

(Execution)  

Response  

Note:  
ICRW can not receive the data from HOST, at the time of the writing of EEPROM
5. Command/Response

5.1 Command Format (HOST->UNIT)

```
“C” CM PM DATA
```

“C”= 43H.
CM: Command code.
PM: Parameters .
DATA: With data part and without data part .

5.2 Positive response format(ICRW->HOST)

```
“P” CM PM ST1 ST0 DATA
```

“P”= 50H.
In this format CM and PM returns the same values which were received with command transmission.(except for IC card control).
ST0, ST1: Status code.
There are positive responses with data part and without data part.

5.3 Negative response format(ICRW->HOST)

```
“N” CM PM E1 E0 DATA
```

“P”= 4EH
In this format CM and PM returns the same values which were received with command transmission.(except for IC card control).
E1,E0: Error code. There are negative responses with data part and without data part.

Note:
Command must be transmitted more than 5msec after receipt of the response from ICRW.
6. CRT-350 Executed code area (including status code and error code)

6.1 Command list (blue parts indicate the difference with SANKYO 3Q8)

List1 cm: Command code    pm: Parameters

<table>
<thead>
<tr>
<th>Command</th>
<th>cm</th>
<th>pm</th>
<th>Details of operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>30H</td>
<td>30H</td>
<td>If card is inside ICRW, move card to gate (with holding card in front)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31H</td>
<td>If card is inside ICRW, capture card backward (without holding card in rear)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32H</td>
<td>If card is inside ICRW, retain the card inside of ICRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33H</td>
<td>If card is inside ICRW, does not move the card</td>
</tr>
<tr>
<td></td>
<td>34H</td>
<td></td>
<td>Same as 30H. And Retract counter will work</td>
</tr>
<tr>
<td></td>
<td>35H</td>
<td></td>
<td>Same as 31H. And Retract counter will work</td>
</tr>
<tr>
<td></td>
<td>36H</td>
<td></td>
<td>Same as 32H. And Retract counter will work</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40H</td>
<td>Same as 30H, but move card to gate(without holding card in front)</td>
</tr>
</tbody>
</table>

List2 cm: Command code    pm: Parameters

<table>
<thead>
<tr>
<th>Command</th>
<th>cm</th>
<th>pm</th>
<th>Details of operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve (See 8.5)</td>
<td>34H</td>
<td>30H</td>
<td>Retrieve card, which is in Gate position</td>
</tr>
<tr>
<td></td>
<td>35H</td>
<td></td>
<td>Nonsupport for CRT-350</td>
</tr>
<tr>
<td>Mag-Track Read</td>
<td>36H</td>
<td>30H</td>
<td>Only card movement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31H</td>
<td>ISO Track #1 reads</td>
</tr>
<tr>
<td>Command</td>
<td>cm</td>
<td>pm</td>
<td>Details of operation</td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
<td>----</td>
<td>---------------------</td>
</tr>
<tr>
<td>Revision (See 8.10)</td>
<td>41H</td>
<td>31H</td>
<td>Version of ICRW software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32H</td>
<td>Version of EMV software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33H</td>
<td>EMV Type Approval Reference number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34H</td>
<td>GIE CB Type Approval Reference number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35H</td>
<td>Same as SANKYO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36H</td>
<td>Version of CPU card (7816) controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41H</td>
<td>Version of memory controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td>42H</td>
<td>Version of M1 RF card controller</td>
</tr>
<tr>
<td>Retract Counter</td>
<td>43H</td>
<td>30H</td>
<td>Retract counter read</td>
</tr>
</tbody>
</table>
List 4 cm: Command code  pm: Parameters

<table>
<thead>
<tr>
<th>Command</th>
<th>cm</th>
<th>pm</th>
<th>Details of operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siemens Memory Card Control</td>
<td>52H</td>
<td>30H</td>
<td>To activate Siemens memory card.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31H</td>
<td>To deactivate Siemens memory card.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32H</td>
<td>To inquire Siemens memory card status.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33H</td>
<td>To exchange data between SLE4442.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34H</td>
<td>To exchange data between SLE4428.</td>
</tr>
<tr>
<td>Monitoring For removal</td>
<td>54H</td>
<td>30H</td>
<td>To monitoring for card removal.</td>
</tr>
<tr>
<td>(See 8.14)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple magnetic track read</td>
<td>56H</td>
<td>31H</td>
<td>Transmit read data ISO #1</td>
</tr>
<tr>
<td>(See 8.6.3)</td>
<td></td>
<td>32H</td>
<td>Transmit read data ISO #2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33H</td>
<td>Transmit read data ISO #3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34H</td>
<td>Transmit read data ISO #1 &amp; ISO #2</td>
</tr>
<tr>
<td>Communication Protocol</td>
<td>Model No.</td>
<td>CRT-350</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>2010/1/19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ver.</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>15/155</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 35H | Transmit read data ISO #1 & ISO #3 |
| 36H | Transmit read data ISO #2 & ISO #3 |
| 37H | Transmit read data ISO #1 & ISO #2 & ISO #3 |
| 51H | Transmit read data ISO #1 by binary |
| 52H | Transmit read data ISO #2 by binary |
| 53H | Transmit read data ISO #3 by binary |

| 6CH | Special inject/eject card control (See 8.3.3) |
| 30H | Accept card by time-restriction switch mode |
| 32H | Accept card by time-restriction magnetic card mode |
| 40H | Eject card to gate, and time-restriction capture card |

| 80H | Operation for red indicator |
| 31H | Operation for red indicator |
| 32H | Operation for green indicator |

| 90H | Automatically Detect Card Type (See 8.17) |
| 30H | Automatically detect IC card type |
| 31H | Automatically detect RF card type |

| 91H | Memory Card operation (See annex II) |
| 30H | To reset by power on |
| 31H | Power off |
| 32H | Inquire card status |
| 33H | Operation of memory card |
### List5 cm: Command code  pm: Parameters

<table>
<thead>
<tr>
<th>Command</th>
<th>cm</th>
<th>pm</th>
<th>Details of operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 RF card operation (See annex III)</td>
<td>92H</td>
<td>30H</td>
<td>Activate/search for card</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31H</td>
<td>Power off</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32H</td>
<td>Inquire card status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33H</td>
<td>Operation of MIFARE ONE card</td>
</tr>
<tr>
<td>Lifetime of Wearing parts (See 8.19)</td>
<td>A1H</td>
<td>30H</td>
<td>Read the times of passing MAG head, opening gate, IC contact</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31H</td>
<td>Initialize the times of passing MAG head, opening gate, IC contact</td>
</tr>
<tr>
<td>Write/Read SN (See 8.20)</td>
<td>A2H</td>
<td>30H</td>
<td>Read ICRW ‘s SN</td>
</tr>
<tr>
<td>Read Configured characters (See 8.21)</td>
<td>A3H</td>
<td>30H</td>
<td>Read the configured characters of ICRW</td>
</tr>
</tbody>
</table>
6.2 Status Code

ST1, ST0: ICRW status code

<table>
<thead>
<tr>
<th>ST1 &amp; ST0</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>“00”</td>
<td>No card detected within ICRW (including card gate)</td>
</tr>
<tr>
<td>“01”</td>
<td>Card locates at card Gate</td>
</tr>
<tr>
<td>“02”</td>
<td>Card locates inside ICRW (Transport)</td>
</tr>
</tbody>
</table>

6.3 Error Code

List 1 E1, E0: error code

<table>
<thead>
<tr>
<th>E1 &amp; E2</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>“00”</td>
<td>A given command code is unidentified</td>
</tr>
<tr>
<td>“01”</td>
<td>Parameter is not correct</td>
</tr>
<tr>
<td>“02”</td>
<td>Command execution is impossible</td>
</tr>
<tr>
<td>“03”</td>
<td></td>
</tr>
<tr>
<td>“04”</td>
<td>Command data error</td>
</tr>
<tr>
<td>“05” - “09”</td>
<td></td>
</tr>
<tr>
<td>“10”</td>
<td>Card jam</td>
</tr>
<tr>
<td>“11”</td>
<td>Shutter failure</td>
</tr>
<tr>
<td>“12”</td>
<td>Sensor failure of PD1, PD2, PD3, PDI (Card remains inside)</td>
</tr>
<tr>
<td>“13”</td>
<td>Irregular card length (LONG)</td>
</tr>
<tr>
<td>“14”</td>
<td>Irregular card length (SHORT)</td>
</tr>
<tr>
<td>“15”</td>
<td>EEPROM error</td>
</tr>
<tr>
<td>“16”</td>
<td>The card was moved forcibly.</td>
</tr>
<tr>
<td>“17”</td>
<td>Jam error at retrieve</td>
</tr>
<tr>
<td>“18”</td>
<td>SW1 or SW2 error</td>
</tr>
<tr>
<td>“19”</td>
<td>Card was not inserted from the rear</td>
</tr>
</tbody>
</table>

List 2 E1, E0: error code
<table>
<thead>
<tr>
<th>E1 &amp; E2</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>“20”</td>
<td>Read Error (Parity error)</td>
</tr>
<tr>
<td>“21”</td>
<td>Read Error</td>
</tr>
<tr>
<td>“22”</td>
<td>Write Error</td>
</tr>
<tr>
<td>“23”</td>
<td>Read Error (Only SS-ES-LRC)</td>
</tr>
<tr>
<td>“24”</td>
<td>Read Error (no code and/or no magnetic stripe)</td>
</tr>
<tr>
<td>“25”</td>
<td>Write Verify Error (Quality error)</td>
</tr>
<tr>
<td>“26”</td>
<td>Read Error (No SS)</td>
</tr>
<tr>
<td>“27”</td>
<td>Read Error (No ES)</td>
</tr>
<tr>
<td>“28”</td>
<td>Read Error (LRC error)</td>
</tr>
<tr>
<td>“29”</td>
<td>Write Verify Error (Data discordance)</td>
</tr>
<tr>
<td>“30”</td>
<td>Power Down</td>
</tr>
<tr>
<td>“31”</td>
<td>DSR signal was turned to OFF</td>
</tr>
<tr>
<td>“32”–“39”</td>
<td>--</td>
</tr>
<tr>
<td>“40”</td>
<td>Card was pulled out during capture</td>
</tr>
<tr>
<td>“41”</td>
<td>Failure at IC contact solenoid or sensor IC</td>
</tr>
<tr>
<td>“42”</td>
<td>--</td>
</tr>
<tr>
<td>“43”</td>
<td>Card could not be to IC contact position</td>
</tr>
<tr>
<td>“44”</td>
<td>--</td>
</tr>
<tr>
<td>“45”</td>
<td>ICRW ejected the card failure</td>
</tr>
<tr>
<td>“46”</td>
<td>The ejected card has not been withdrawn until the specified time</td>
</tr>
<tr>
<td>“47”–“49”</td>
<td>--</td>
</tr>
<tr>
<td>“50”</td>
<td>Retract counter overflow</td>
</tr>
</tbody>
</table>

List 3 E1, E0: error code
<table>
<thead>
<tr>
<th>E1 &amp; E2</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>“51”</td>
<td>Motor error</td>
</tr>
<tr>
<td>“52”</td>
<td>--</td>
</tr>
<tr>
<td>“53”</td>
<td>Digital decode read error</td>
</tr>
<tr>
<td>“54”-“59”</td>
<td>--</td>
</tr>
<tr>
<td>“60”</td>
<td>Abnormal condition was found on the power-line (VCC) of IC card</td>
</tr>
<tr>
<td>“61”</td>
<td>ATR error</td>
</tr>
<tr>
<td>“62”</td>
<td>The specified protocol does not agree with that of IC card</td>
</tr>
<tr>
<td>“63”</td>
<td>IC card communication error (IC card does not respond)</td>
</tr>
<tr>
<td>“64”</td>
<td>IC card communication error (Other than “63”)</td>
</tr>
<tr>
<td>“65”</td>
<td>HOST sends command for CPU card communication before receiving ATR</td>
</tr>
<tr>
<td>“66”</td>
<td>Tried to communicate with IC card not supported in ICRW</td>
</tr>
<tr>
<td>“67-“68”</td>
<td>--</td>
</tr>
<tr>
<td>“69”</td>
<td>Tried to communicate with IC card not supported in Protocol EMV2000</td>
</tr>
<tr>
<td>“80”</td>
<td>When write-check by binary, data in the track is zero</td>
</tr>
<tr>
<td>“81”</td>
<td>When write-check by binary, the check data is error</td>
</tr>
<tr>
<td>“90”</td>
<td>Unknown type card</td>
</tr>
<tr>
<td>“B0”</td>
<td>Not received Initialize command</td>
</tr>
</tbody>
</table>
7. Structure and Location Schematic of CRT-350

- Structure schematic

- Location schematic
8. Detailed Description of CRT-350 Commands

8.1 Initialize command

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>30H</th>
<th>PM</th>
<th>33H</th>
<th>32H</th>
<th>34H</th>
<th>30H</th>
<th>FM</th>
<th>PD</th>
<th>WV</th>
<th>SH</th>
<th>DS</th>
<th>TY</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>30H</td>
<td>PM</td>
<td>ST1</td>
<td>ST0</td>
<td>Type recognizing code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>30H</td>
<td>PM</td>
<td>E1</td>
<td>E0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is the first command executed whenever power is turned on, or any else is not executed, it can be done many times after successfully.
ICRW determines text configuration and recognizes baud rate automatically upon receipt of this command.
ICRW is initialized in Disable mode that card is not accepted by this command.
Use this command to recover from mechanical error and power down mode.

"*" These are the parameters to specify the tracks and the register in the ASCII encoder circuit is assigned for each track format as ISO standard
ISO-1track : 33H indicates IATA format
ISO-2track : 32H indicates ABA format
ISO-3track : 34H indicates MINTS format
ISO-4track : Special track, 30H indicates no

PM : ICRW disposes a card which is held inside ICRW.
If no card in it, then ICRW will drive motor lightly
If card in it, ICRW moves card to rear(with holding card), then test intake as SH, and executes command below:
=30H Move the card to Gate portion(with holding card)
(Not to let the motor rotate, if ICRW is initialized with SH ="1" and if the card covers SW1 only)
=31H Capture card to rear side of ICRW
=32H If card is inside ICRW, detain the card inside of ICRW.
=33H Does not move the card.
=34H Same as pm=30H, and Retract counter will work
=35H Same as pm=31H, and Retract counter will work.
=36H Same as pm=32H, and Retract counter will work.
=40H Same as pm=30H, just retract card to intake(without holding card)

FM: Not use. Always 30H.
PD: At Power down, ICRW controls Card
   =30H: ICRW does card eject(gate without card holding)
   =31H: Eject is unavailable during MAG-stripe write operation, but else is available
   =32H: Eject is unavailable in any case (if..... )
   =33H: Any ICRW function to have been completed, then offer the card back to the customer and if the card is not taken by the customer after 30 seconds, capture the card.

WV: Write command handling at power down.
   =30H: Unit stops the operation against power failure occurred during Mag-stripe WRITE operation.
   =31H: Unit continues operation neglecting power failure occurred during Mag-stripe WRITE operation.

SH: Testing shutter operations (open/close).
   =30H: Unit tests open/close test of shutter when Initialize command is executed.
   =31H: Unit does not test open/close test of shutter when Initialize command is executed.
If ICRW does not have shutter, then it will neglect this parameter test no matter what any value

Ds: Monitoring transmission control signal DSR of RS232(E.g.: DSR=0FF, Stop exchange).
   =30H: Only at POWER ON, line control signal DSR is checked.
   =31H: Always, line control signal is checked. If the status is detected as OFF, The card staying inside of the transport is ejected to the entrance gate(without card holding). All the SAM are deactivated.
   =32H: Always, line control signal is checked. If the status is detected as OFF, The card staying inside of the transport is ejected to the entrance gate(with card holding). All the SAM are deactivated.

TY: Reader type recognition code.
   30H: No data is contained in the response.
   31H: Response includes type-recognizing code.

<table>
<thead>
<tr>
<th>Type recognizing code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO#1 ISO#2 ISO#3 &quot;0&quot; ICT/MCT High-Co/Low-Co RF SAM6 SAM7 SAM8</td>
</tr>
<tr>
<td>SAM1 SAM2 SAM3 SAM4 SAM5 &quot;0&quot; -- -- -- -- --</td>
</tr>
</tbody>
</table>

RF: Addition of CRT-350

SAM n: all SAM cards is option, and can only be executed in PSAM

Value of recognizing code:
<table>
<thead>
<tr>
<th>Item</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| ISO#1,ISO#2,ISO#3 | "0"– None  
|              | "1"– Read only  
|              | "2"– Read/write                                                              |
| ICT/MCT      | "0" – No ICC operation board  
|              | "1" – Operate ICC                                                           |
| High-Co/Lo-Co | "0"– Low-Co operation  
|              | "1"– High-Co operation                                                      |
| RF           | "0"– No RF operation board  
|              | "1"– Operate RF                                                             |
| SAM n        | "0"– Socket is absence.  
|              | "1"– Socket is presence.  
|              | "2"– SAM chip is presence                                                   |

**CP**: Error status set against card pulled out when CAPTURE command is executed.  
=30H: Do not set any error status  
=31H: Set error status to be “40”

**Notes:**  
(1) DSR signal is not monitored during execution of command.  
(2) WV, SH, Ds, TY and CP are not always specified.  
When any of them is not specified, it will have "0" as default value.  
(3) When power failure occurs at the same timing of DSR OFF, power failure handling routine has priority.
8.2 Status request command

8.2.1 Require status

Command

```
"C" 31H PM
```

Positive response

```
"P" 31H PM ST1 ST0
```

Negative response

```
"N" 31H PM E1 E0
```

PM = 30H: Report current status, without SE
PM = 31H: Response is returned in form of SE

```
SE= 0 1 0 X X X X X
```

SW1** 1: Card exit, 0: Card does not exist
PD1 1: Card exists, 0: Card does not exist
PD2 1: Card exists, 0: Card does not exist
PD3 1: Card exists, 0: Card does not exist

SW2

1: Shutter is open or card locate the shutter
0: Shutter is closed.
Expansion Status Request Command

Command

```
"C" 31H PM
```

Positive response

```
"P" 31H PM ST1 ST0 SE1 SE0 SE2 SE3
```

Negative response

```
"N" 31H PM E1 E0
```

PM=32H: Response is returned in form of SEn with the status information obtained.
PM=33H: Response is returned in form of SEn (the same data that case of PM=32H.) with the status information obtained. while mechanical error occurred.

```
b7
SE1= 0 1 0 0 0 X X X
     (Binary)

PHD 1: MAG signal exists, 0: MAG signal does not exist
PD1 1: Card exists, 0: Card does not exist
ICD 1: Card exists, 0: Card does not exist
```

* PHD status keeps 0 and has no meaning after card insertion (Ie : st.1, st.0 = 2)

```
b7
SE0= 0 1 0 X X X X X
     (Binary)

PD3 1: Card exists, 0: Card does not exist
PD2 1: Card exists, 0: Card does not exist
PD1 1: Card exists, 0: Card does not exist
SW2 1: Shutter is open or card locates beneath the shutter
      0: Shutter is closed.
SW1** 1: Card exit, 0: Card does not exist
```
MC2=1 denotes it detects the magnetic flux signal after SW1=0
**Require sensor status of ICRW**

**Command**

```
"C"  31H  PM
```

**Positive response**

```
"P"  31H  PM  ST1  ST0  LN
```

**Negative response**

```
"N"  31H  PM  E1  E0
```

PM=40H  Return the status of ICRW in the form of LN(1byte)

**LSN format:**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>S6</td>
<td>S5</td>
<td>S4</td>
<td>S3</td>
<td>S2</td>
<td>S1</td>
</tr>
</tbody>
</table>

S1-- S6(Status of infrared sensor)  S(1…6)=0  Do not detect card

S(1…6)=0  Detect card

**8.2.2  Require intake status of ICRW**

**Command**

```
"C"  31H  PM
```

**Positive response**

```
"P"  31H  PM  ST1  ST0  Intake type (P)  PS1  PS2  PS3  PS4
```

**Negative response**

```
"N"  31H  PM  E1  E0
```
PM=41H Return the intake status of ICRW
Intake type (P): =0x30 no intake , S1/S2/S3/S4=0x3F, denote no meaning
=0x3D ‘D’ type intake, S3/S4=0x3F, denote no meaning
=0x3E ‘E’ type intake
PS1 Shutter echo =0x30 Shutter close
=0x31 Shutter open

PS2 status of SW1 =0x30 Detect the signal of card inserting shutter
=0x31 Do not detect the signal of card inserting shutter

PS3 status of Anti-theft pin =0x30 Anti-theft pin activation
=0x31 Anti-theft pin deactivation
PS4 set of vibrating intake =0x30 Vibrating intake activation
=0x31 Vibration intake deactivation
8.3 Entry command

8.3.1 Entry for once

Command

\[
\begin{array}{c|c|c|c}
\text{"C"} & \text{32H} & \text{PM} \\
\end{array}
\]

Positive response

\[
\begin{array}{c|c|c|c|c}
\text{"P"} & \text{32H} & \text{PM} & \text{ST1} & \text{ST0} \\
\end{array}
\]

Negative response

\[
\begin{array}{c|c|c|c|c}
\text{"N"} & \text{32H} & \text{PM} & \text{E1} & \text{E0} \\
\end{array}
\]

PM=30H: To accept the card with switch mode (only detect the status of intake and ICRW reality sensor S1, that is accepting any type card)

(This can be executed by the ICRW without magnetic flux shutter)

PM=31H: To accept the card with magnetic card mode (detect the intake switch and Pre-read magnetic head)

This is to accept to carry the card inside ICRW, namely detects whether card is inserted or not by monitoring intake or ICRW reality sensor S1, This command doesn't allow ICRW to send response to HOST until ICRW completes to carry the card inside ICRW.

If ICRW can not move the card on the way of carrying it, ICRW sends jam error"10" to HOST.

If another card already stays inside ICRW, ICRW sends error "02" to HOST.

Send DLE+EOT from HOST in order to cancel this command.

This is executed right in the status of prohibited accepting card, if card inside ICRW or enable for continued accepting card, then ICRW sends error "02" to HOST.

When card inserting, ICRW will make an operation of reading magnetic, and the reading data is stored in register instead of sending to HOST

(even if the data is error, ICRW also sends the positive response, since the card is moved successfully)

PM=32H: To accept the card from rear side of ICRW with time control

When receiving the command with this parameter, ICRW detects the rear sensor S6, if a card insertion allows ICRW to carry the card inside ICRW(in front of magnetic head).
When ICRW can't detect a card insertion for 10sec, ICRW stops the motor and sends error "19" to HOST.
If ICRW can not move the card on the way of carrying it, ICRW sends jam error"10" to HOST.
If another card already stays inside ICRW, ICRW sends error "02" to HOST.

8.3.2 Entry for many times

<table>
<thead>
<tr>
<th>Command</th>
<th>Positive response</th>
<th>Negative response</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>P</td>
<td>N</td>
</tr>
</tbody>
</table>

This is the same as the command of 32H in entry for once ,but it mustn’t send the command of entry for once to ICRW for many times, which is similar with the card inject control in CRT-310

<table>
<thead>
<tr>
<th>PM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30H</td>
<td>To accept card by switch mode( for any standard card)</td>
</tr>
<tr>
<td>31H</td>
<td>Prohibit accepting card</td>
</tr>
<tr>
<td>32H</td>
<td>To accept card by magnetic card mode(for magnetic card ,and right direction)</td>
</tr>
<tr>
<td>40H</td>
<td>To accept card from rear</td>
</tr>
</tbody>
</table>

Note:
- After sending a command of accepting card from rear , HOST can send a command of prohibiting inserting card from gate in the form of PM=31H
- ICRW prohibits card insertion in the stations below:
  - Re-power ICRW
  - Executing initialize command
  - Card inside ICRW
  - DS="1" or “2” as initialize ICRW, and then ICRW detect DSR=OFF
8.3.3 card in/out Jitter function setting

Command

```
“C” 3AH 58H mode
```

Positive response

```
“C” 3AH 58H st1 st0
```

Negative response

```
“N” 3AH 58H e1 e0
```

Mode: Card in/out jitter function setting

- **30H**: No set jitter function when card in (Default)
- **31H**: Set jitter function when card in
- **34H**: Set Anti-fishing function when card in, the jittering time in this mode is longer than that in mode=31H

Note:

1. This command is to set the jitter function when card in/out
2. When jitter function is set with the enable card in command (“C20” “C21” “C:0” “C:2”) the jitter function will start just when it detect magnetic card signal.
3. Jitter function will be changed by this command, but will not be changed by reset command
4. Jitter function will not start when reset with power on.

Example:

1. “C:X1” (Start anti-fishing function)
2. “C:2” (Enable card-in from front by magnetic card)
3. Card insertion (Insert card)
8.3.4 Special command of jittering card in/out

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>6CH</th>
<th>PM</th>
<th>TM10</th>
<th>TM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>6CH</td>
<td>PM</td>
<td>ST1</td>
<td>ST0</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>6CH</td>
<td>PM</td>
<td>E1</td>
<td>E0</td>
</tr>
</tbody>
</table>

**Inject card:**
This is similar with the description in 8.3.1, but just the waiting time is controlled by parameters, and detects the status. SE1/SE0/SE2/SE3 in this command is the same as response in Expansion Status Request Command of 8.2.1

- PM=30H: To accept card by switch mode (for any standard card)
- PM=32H: To accept card by magnetic card mode (for magnetic card, and right direction)

If not accept card during the appointed time, ICRW will send error "46" to HOST.

If error happens to the way of moving card, ICRW sends relevant error code to HOST.

ICRW read magnetic card automatically during carrying card inside it, and stores the data in RAM.

**Eject card:**

- PM=40H: retract card to gate with holding card, and wait for taken it.

If no card inside ICRW, ICRW sends error "02" to HOST.

If card inside ICRW, ICRW retract card to gate with holding card to wait for taken it.

If card in intake, ICRW do not move card to wait for taken it, and then sends status to HOST.

The response is the same as 8.2.2, for example if not accept card during the appointed time, ICRW will send error "46" to HOST.

**Time control**

Controlled by TM n, the unit is sec., TM10 denotes tens-digit, TM1 denotes single-digit TM10,TM1="00"—"99", TM10,TM1="00" denotes set time is 0.1sec., TM10,TM1="01" denotes 1sec., TM10,TM1="15" denotes 15 sec.. the default value of TM10,TM1 is infinite.

For example: set value: 15 sec., then TM10=31H or "1", and TM1=35H or "5"

The max value is 99 sec., if overtime errors, then ICRW send error "46" to HOST.

This timer controller is not definition.
8.4 Carry card command

8.4.1 Carry card

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>33H</th>
<th>PM</th>
<th>TM100</th>
<th>TM10</th>
<th>TM1</th>
</tr>
</thead>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;P&quot;</th>
<th>33H</th>
<th>PM</th>
<th>ST1</th>
<th>ST0</th>
</tr>
</thead>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;N&quot;</th>
<th>33H</th>
<th>PM</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
</table>

If a card doesn't stay inside ICRW, ICRW sends error "02" to HOST
PM=30H: Eject
This moves card to Gate portion(with holding card), if with time parameter, then moves card according for it.
When card reaches to Gate, the response is sent.
Time parameter is available just in this PM, the moving distance is made by timer controller(conversion ), unit is ms, TM100 denotes hundred-digit, TM10 denotes tens-digit, TM1 denotes single-digital ,the max value is 20 ms, 1ms is about 0.19mm, once executing this command ,then the card maybe disengage off idler wheel to stop moving card

PM=31H: Capture
When ICRW is initialized with PM=34H-36H, retract counter is increased by one as card is retracted successfully
When capture command is received against retract counter value 99,error "50" is replied in negative response.

PM=32H: Set MM
MM is a special position, it is not protrusive from card injecting port, just reaches the card injecting switch.
8.4.2 Expansion Carry card

Command

```
"C" 33H 40H PL
```

Positive response

```
"P" 33H PM ST1 ST0
```

Negative response

```
"N" 33H PM E1 E0
```

This command makes card move to its any standard position:

- **PL=30H** This moves card to Gate portion (without holding card)
- **PL=31H** This moves card to Gate portion (with holding card)
- **PL=32H** This moves card to in front of MAG head portion to operate RF card
- **PL=33H** This moves card to IC card ready position and set the IC contact, and then to carry card to IC operation position to operate IC card
- **PL=34H** This moves card to in back of MAG head portion (without reading/writing card, just moves card)
- **PL=35H** This moves card to rear portion (with holding card)
- **PL=36H** This moves card to rear portion (without holding card)
- **PL=39H** Retract abnormal card (short/long) to the rear position of ICRW, cancel the execution initialized for counter, there is an assistant for short card by inserting a normal card in card port.

   It is used for clearing ICRW or resolving malfunction
8.5 Retrieve command

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>34H</td>
<td>40H</td>
<td></td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;p&quot;</td>
<td>34H</td>
<td>30H</td>
<td>ST1</td>
<td>ST0</td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td>34H</td>
<td>30H</td>
<td>E1</td>
<td>E0</td>
</tr>
</tbody>
</table>

Move card (stay in Gate portion) into inside ICRW (in the back of MAG. head).
While the card is carried inside ICRW, data on the MAG-stripe is read to data buffer by ICRW without sending to HOST (Even if read error occurs, ICRW sends positive response to HOST upon completion of carrying the card inside ICRW.)
8.6 MAG-Track Read operation command

There are three modes to read MAG-track, commands of which are 36H, 44H, 56H

8.6.1 MAG-Track Read command

<table>
<thead>
<tr>
<th>Command</th>
<th>“C”</th>
<th>36H</th>
<th>PM</th>
</tr>
</thead>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>“P”</th>
<th>36H</th>
<th>PM</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>“N”</th>
<th>36H</th>
<th>PM</th>
<th>E1</th>
<th>E0</th>
<th>DATA</th>
</tr>
</thead>
</table>

PM=30H: Card movements only within ICRW.
Wherever card being, carry card to MAG head, no reading, and stop in front of MAG head or in back of it.

PM=31H: read data on ISO Track #1
PM=32H: read data on ISO Track #2
PM=33H: read data on ISO Track #3

The command with the above parameters allows ICRW to transmit the data of buffer, which have been normally read during the card acceptance. When Read Error occurs, ICRW makes retrying for 5 times until successfully reading. When Read Error still occur, ICRW sends negative response. In case of card jamming, ICRW sends negative response too. Data on the MAG-stripe is SS-ES-LRC, or blank, then ICRW sends negative response without retrieve.

Data format of ISO ASCII code:
-Track#1(IATA): 79 characters max. (6bits+1parity) e.g. b0, b1, b2, b3, b4, b5. P
-Track#2(ABA): 40 characters max. (4bits+1 parity) e.g. b0, b1, b2 b3. P
-Track#3(MINTS): 107 characters max. (4bits+1 parity) e.g. b0, b1, b2 b3. P

For examples:

<table>
<thead>
<tr>
<th>ISO Track #1</th>
<th>ISO Track #2, #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>bit</td>
</tr>
<tr>
<td>data=0</td>
<td>data=0</td>
</tr>
<tr>
<td>data=A</td>
<td>data=9</td>
</tr>
</tbody>
</table>

PM=33H: All three tracks simultaneous read and transmit.
Transmit Format:

<table>
<thead>
<tr>
<th>Track 1 Data</th>
<th>7EH</th>
<th>Track 2 Data</th>
<th>7EH</th>
<th>Track 3 Data</th>
</tr>
</thead>
</table>

When one of the tracks is not being data, its data area becomes blank, for example, the sent data of ISO-Track#2 without data is shown as below:

<table>
<thead>
<tr>
<th>Track 1 Data</th>
<th>7EH</th>
<th>7EH</th>
<th>Track 3 Data</th>
</tr>
</thead>
</table>

When one of the tracks is not read, its data area becomes error code, error: "20", "23","24","26","27" or "28"
PM=36H: This clears out all the stored data for reading and writing.
ICRW doesn't carry the card.
PM=37H: This inspects whether data exist or not in ICRW buffer
ICRW doesn't carry the card.
Transmit Format:

<table>
<thead>
<tr>
<th>ISO#1</th>
<th>ISO#2</th>
<th>ISO#3</th>
</tr>
</thead>
</table>

ISO#1: 30H: ISO #1 is not encoded. 31H: ISO #1 is encoded.
ISO#2: 30H: ISO #2 is not encoded. 31H: ISO #2 is encoded.
ISO#3: 30H: ISO #3 is not encoded. 31H: ISO #3 is encoded.
PM=39H: read data on ISO Track #1 by another way
PM=3AH: read data on ISO Track #2 by another way
PM=3BH: read data on ISO Track #3 by another way
The above parameters send MAG-track data to ICRW buffer and not load, and differ from pm=31H, 32H and 33H in the following contents.
1) This parameter transports card only one pass, even thought the read error has occurred. In this case, ICRW sends negative response.
2) If the parity error occurs, the ICRW tries to send the data row before the error portion. This partial readied data is concatenated the negative response. If SS is not detected, ICRW doesn't read data.

PM=51H: ICRW do not move card ,but sends the ISO #1 data read by binary mode to HOST
PM=52H: ICRW do not move card ,but sends the ISO #2 data read by binary mode to HOST
PM=53H: ICRW do not move card ,but sends the ISO #3 data read by binary mode to HOST

Binary read card differs from others in the sent data that it is not encoded and not checked by ASCII code, which ignores error or right, and which is in the form of converted ASCII code by one ASCII including 4 bits binary code. There are plenty of pre-load and suffix-load zeros in these sent data, so ICRW will ignores these zeros during sending data
For example:  data in a track: (HEX) = ‘0000 0000 0011 0111 1111 0000 0000 0011 0111
1111 0000 0000’

data packet sent to HOST: 0x33 0x37 0x3F 0x30 0x30 0x33 0x37 0x3F
If there is no data in a track, then the sent data is 0x30
8.6.2 MAG-Track special read command

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>44H PM</td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>PM</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td>44H PM</td>
<td>ST1</td>
<td>ST0</td>
<td>DATA</td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>PM</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td>44H PM</td>
<td>ST1</td>
<td>ST0</td>
<td>DATA</td>
</tr>
</tbody>
</table>

This is a special read MAG-track command in SANKYO, which is reserved in CRT-350.

PM=31H: read data on ISO Track #1
PM=32H: read data on ISO Track #2
PM=33H: read data on ISO Track #3

Carry and read card, and then send data to HOST. If errors occur in carrying and reading card, then ICRW sends negative response. Data on the MAG-stripe is SS-ES-LRC, or blank, then ICRW sends negative response without retrieve.

Data format in responded data is familiar of PM=31H/32H/33H in the command of 36H, but this will send data to HOST.
8.6.3 Multi-MAG-Track read command

Command  
"C"  56H  PM

Positive response  
"P"  564H  PM  ST1  ST0  DATA

Negative response  
"N"  56H  PM  ST1  ST0  DATA

PM=31H:  read data on ISO Track #1  
PM=32H:  read data on ISO Track #2  
PM=33H:  read data on ISO Track #3  
PM=34H:  read data on ISO Track #1, ISO Track #2  
PM=35H:  read data on ISO Track #1, ISO Track #3  
PM=36H:  read data on ISO Track #2, ISO Track #3  
PM=37H:  read data on ISO Track #1, ISO Track #2, ISO Track #3

The command with the above parameters allows ICRW to transmit the data of buffer, which have been normally read during the card acceptance. When Read Error occurs, ICRW makes retrying for 5 times until successfully reading. When Read Error still occurs, ICRW sends negative response and error code is 21. In case of card carrying, ICRW sends negative response too. Data on the MAG-stripe is SS-ES-LRC, or blank, then ICRW sends negative response without retrieve.

Data format of ISO ASCII code:  
-Track#1(IATA): 79 characters max. (6bits+1parity) e.g. b0, b1, b2, b3, b4, b5, P  
-Track#2(ABA): 40 characters max. (4bits+1 parity) e.g. b0, b1, b2 b3, P  
-Track#3(MINTS): 107 characters max. (4bits+1 parity) e.g. b0, b1, b2 b3, P

For example

<table>
<thead>
<tr>
<th>ISO Track #1</th>
<th>ISO Track #2, #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>bit</td>
</tr>
<tr>
<td>data=0</td>
<td>data=0</td>
</tr>
<tr>
<td>data=A</td>
<td>data=9</td>
</tr>
</tbody>
</table>

Response data format: ISO#1 data+7EH+ ISO#2 data+7EH+ ISO#3 data

ISO# n data:
- Positive read: “P”+ track data (ASCII code excludes SS-ES-LRC)
- Negative read: “N2X” , “2X” error code ( “20”“23”“24”“26”“27”“28” )
Positive response:
PM=31H: "PV102" + "P" + ISO #1 data
PM=32H: "PV202" + "P" + ISO #2 data
PM=33H: "PV302" + "P" + ISO #3 data
PM=34H: "PV402" + "P" + ISO #1 data + 7EH + "P" + ISO #2 data
PM=35H: "PV502" + "P" + ISO #1 data + 7EH + "P" + ISO #3 data
PM=36H: "PV602" + "P" + ISO #2 data + 7EH + "P" + ISO #3 data
PM=37H: "PV702" + "P" + ISO #1 data + 7EH + "P" + ISO #2 data + 7EH + "P" + ISO #3 data

All negative response:
E1, E0: "21"
E3, E2: error code of ISO#1
E5, E4: error code of ISO#2
E7, E6: error code of ISO#3
PM=31H: "NV121" + "N" + E3, E2
PM=32H: "NV221" + "N" + E5, E4
PM=33H: "NV321" + "N" + E7, E6
PM=34H: "NV421" + "N" + E3, E2 + 7EH + "N" + E5, E4
PM=35H: "NV521" + "N" + E3, E2 + 7EH + "N" + E7, E6
PM=36H: "NV621" + "N" + E5, E4 + 7EH + "N" + E7, E6
PM=37H: "NV721" + "N" + E3, E2 + 7EH + "N" + E5, E4 + 7EH + "N" + E7, E6

Part negative response:
E1, E0: "21"
It is the same as the all negative response except "P" + ISO#N exists in the positive read in a track.
For example: read for ISO#1 and ISO#2, and positive read in #2, negative read in #3
PM=36H: "NV621" + "P" + ISO #2 data + 7EH + "N" + E7, E6

PM=51H: ICRW carries card (positive data then do not carry card) to read track, and sends ISO#1 binary data in buffer to HOST
PM=52H: ICRW carries card (positive data then do not carry card) to read track, and sends ISO#2 binary data in buffer to HOST
PM=53H: ICRW carries card (positive data then do not carry card) to read track, and sends ISO#3 binary data in buffer to HOST

Binary read card differs from others in the sent data that it is not encoded and not checked by ASCII code, which ignores error or right, and which is in the form of converted ASCII code
by one ASCII including 4 bits binary code. There are plenty of pre-load and suffix-load zeros in these sent data, so ICRW will ignores these zeros during sending data.

For example: slice data in a track: (HEX) = ‘0000 0000 0011 0111 1111 0000 0000 0011 0111 1111 0000 0000’

slice data packet sent to HOST: 0x33 0x37 0x3F 0x30 0x30 0x33 0x37 0x3F
If there is no data in a track, then the sent data is 0x30
8.7 MAG-Track write operation command

8.7.1 MAG-Track write command(1)

Command | "C" | 37H | PM | DATA |
--- | --- | --- | --- | --- |
Positive response | "P" | 37H | PM | ST1 | ST0 | Result#1 | Result#2 | Result#3 |
Negative response | "N" | 37H | PM | E1 | E0 | Result#1 | Result#2 | Result#3 |

PM=31H: Write data on ISO Track #1
PM=32H: Write data on ISO Track #2
PM=33H: Write data on ISO Track #3

Transmit data part by ASCII code to ICRW. It is not necessary to add SS, ES, and LRC.

As for the write of data, a card move is performed in the state of reaching the front of MAG head, and then ICRW moves card through MAG head backward, and makes verify write (done by read operation), the verify write is performed only time in back read.

For example:

If card is in back of ICRW with card holding, ICRW carries it to front position of MAG head, then moves it through MAG head to rear to write card, and makes the verify read by the reverse direction.

If writing is done correctly, ICRW sends positive response with three results including verify read (VF=30H).

When write error occur, ICRW retries 2 times.

When write error still occurs, ICRW sends negative response with three results including verify read, in this case, error code E1 and E0 are “22”.

In case of card jamming, ICRW sends negative response.

Result structure:

Good write: “P”+“0”+ VF (3 characters)
Bad write: “N”+“2X” (“20”“24”“25”“26”“27”“28”“29”)
No write*: “X”+“00"

(* This track is not designated by the parameter.)

Ex)
(1)#1 good
Positive response “P0”+ “0” + 7EH +“X00”+ 7EH + “X00”
(2) #2 bad
Negative response  “N”+ “2X” + 7EH + “X00” + 7EH + “X00”  
(Error code E1 and E0 : “22”)
8.7.2 MAG-Track write command(2)

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>37H</th>
<th>PM</th>
<th>DATA</th>
</tr>
</thead>
</table>

Positive response

| Positive response | "P" | 37H | PM | ST1 | ST0 |

Negative response

| Negative response | "N" | 37H | PM | E1 | E0 |

PM=36H: Write data to ISO Track #1 buffer
PM=37H: Write data to ISO Track #2 buffer
PM=38H: Write data to ISO Track #3 buffer

This is to write the data to ISO track buffer, but ICRW does not write card.
All the stored data are cleared when card is eject.
All the stored data are cleared when ICRW received initialize command("C00") or clear memory command("C66")
8.7.3 MAG-Track write command(3)

**Command**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
</tr>
<tr>
<td>37H</td>
</tr>
<tr>
<td>39H</td>
</tr>
</tbody>
</table>

**Positive response**

<table>
<thead>
<tr>
<th>Command</th>
<th>37H</th>
<th>39H</th>
<th>ST1</th>
<th>ST0</th>
<th>Result#1</th>
<th>Result#2</th>
<th>Result#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td>37H</td>
<td>39H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Negative response**

<table>
<thead>
<tr>
<th>Command</th>
<th>37H</th>
<th>39H</th>
<th>E1</th>
<th>E0</th>
<th>Result#1</th>
<th>Result#2</th>
<th>Result#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td>37H</td>
<td>39H</td>
<td>E1</td>
<td>E0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PM=39H: All three tracks simultaneous write
ICRW write data to the card from the buffer.
Before writing operation, ICRW need to write data to the buffer.
If there is no data in buffer, ICRW sends back error code"02".
As for the write of data, a card move is performed in the state of reaching the front of MAG head, and then ICRW moves card through MAG head backward, and makes verify write (done by read operation), the verify write is performed only time in back read.

For example:
If card is in back of ICRW with card holding, ICRW carries it to front position of MAG head, then moves it through MAG head to rear to write card, and makes the verify read by the reverse direction.
If writing is done correctly, ICRW sends positive response with three results including verify read(VF=30H)
When write error occurs, ICRW retries 2 times.
When write error still occurs, ICRW sends negative response with three results including verify read, in this case, error code E1 and E0 are “22”.
In case of card jamming, ICRW sends negative response.
Result structure:
Good write: “P”+“0”+ VF (3 characters)
Bad write: “N”+“2X” (“20”“24”“25”“26”“27”“28”“29”)
No write*: “X”+“00”
(* This track is not designated by the parameter.)

Ex)
(1)#1 good, #2 good, #3 good
Positive response “P0”+ “0” + 7EH +“P0”+“0”+ 7EH + “P0”+“0”
(2) #1 good, #2 bad, #3 no write
<table>
<thead>
<tr>
<th>Communication Protocol</th>
<th>Model No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CRT-350</td>
</tr>
<tr>
<td>Date</td>
<td>2010/1/19</td>
</tr>
<tr>
<td>Ver.</td>
<td>3.0</td>
</tr>
<tr>
<td>Page</td>
<td>47/155</td>
</tr>
</tbody>
</table>

Negative response   “P0”+“0”+7EH+“N”+“2X”+7EH+“X00”

(Error code E1 and E0 : “22”)
8.7.4 MAG-Track write command (4)

This is to set mode for write operation
CO:  Set writing current for different coercivity card
CO=30H:  Lo-Co card writing
CO=31H:  Hi-Co card writing
Pay attention to the consistency in the setting mode command for Lo-Co and Hi-Co card.
When writing with Lo-Co ICRW, the data on write track and the neighboring tracks can not be guaranteed.

MODE:  30H and 31H can be used (for compatible with SANKYO), better use 30H
8.7.5 MAG-Track write command(5)

<table>
<thead>
<tr>
<th>Command</th>
<th>Positive response</th>
<th>Negative response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>&quot;P&quot;</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>37H PM DATA</td>
<td>37H PM ST1 ST0</td>
<td>37H PM E1 E0</td>
</tr>
</tbody>
</table>

PM=51H: Write binary data from HOST to ICRW buffer by ASCII code, and then write the data on ISO Track #1
PM=52H: Write binary data from HOST to ICRW buffer by ASCII code, and then write the data on ISO Track #2
PM=53H: Write binary data from HOST to ICRW buffer by ASCII code, and then write the data on ISO Track #3.

All the binary data in any track are in the form of converted ASCII code by one ASCII including 4 bits binary code.

For example: slice data in a track: (HEX) = ‘0011 0111 1111’
slice data packet sent to HOST: 0x33 0x37 0x3F

As for the write of data, a card move is performed in the state of reaching the front of MAG head, and then ICRW moves card through MAG head backward, and makes verify write (done by read operation), the verify write is performed only time in back read. Verify write in binary mode differs from others, ICRW will compare the read binary data from being “1” with the written data bits until comparing all the write data, it is right to pass all the verify, or else it is bad.

Two special error codes for binary verify write:

“80”: do not find the data “1”
“81”: it is bad in the compare

If writing is done correctly, ICRW sends positive response to HOST.
When write error occurs, ICRW retries 2 times (including verify write).
when write error still occurs, ICRW sends negative response with three results including verify read, in this case, error code E1 and E0 are “22”, “80” and “81”
In case of card jamming, ICRW sends negative response.
<table>
<thead>
<tr>
<th>Communication Protocol</th>
<th>Model No.</th>
<th>CRT-350</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>2010/1/19</td>
<td></td>
</tr>
<tr>
<td>Ver.</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>50/155</td>
<td></td>
</tr>
</tbody>
</table>

8.8 Photoelectric sensor detect (No support for CRT-350)
8.9 IC Contact command

<table>
<thead>
<tr>
<th>Command</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>40H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Positive response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
</tr>
<tr>
<td>40H</td>
</tr>
<tr>
<td>PM</td>
</tr>
<tr>
<td>ST1</td>
</tr>
<tr>
<td>ST0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Negative response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>40H</td>
</tr>
<tr>
<td>PM</td>
</tr>
<tr>
<td>E1</td>
</tr>
<tr>
<td>E0</td>
</tr>
</tbody>
</table>

If card does not stay inside ICRW, ICRW sends error code “02” to HOST against receipt of this command

PM=30H: Set IC contact
To carry the card to IC contact position and set the IC contact.
This is familiar with the card move command of moving card to IC position.

When a card cannot be normally set to the position, ICRW makes retrying for 2 times
As a result of re-trial, in the case of an error, ICRW sends a negative response,
If card has reached normal position, then retrieve card.
When receiving initialize command, ICRW release IC contact and performs card movement according to PM of initialize command.

PM=32H: Release IC contact
To release the IC contact.
When IC contact cannot be released, ICRW sends a negative response.
If a card stay inside ICRW, even when IC contact is not set, ICRW sends a positive response to HOST.
### 8.10 Revision read command

**Command**

```
"C" 41H PM
```

**Positive response**

```
"P" 41H PM ST1 ST0 DATA
```

**Negative response**

```
"N" 41H PM E1 E0
```

This is HOST achieves the revisions of software and firmware from ICRW.

<table>
<thead>
<tr>
<th>PM</th>
<th>Meaning</th>
<th>DATA (ASCII)</th>
<th>Length (byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>31H</td>
<td>ICRW Software revision</td>
<td>EX “V20A9211”</td>
<td>8</td>
</tr>
<tr>
<td>32H</td>
<td>EMV(CPU) Software revision</td>
<td>EX “V20A9211”</td>
<td>8</td>
</tr>
<tr>
<td>33H</td>
<td>EMV verify certification</td>
<td>-----</td>
<td>21</td>
</tr>
<tr>
<td>34H</td>
<td>GIE</td>
<td>-----</td>
<td>22</td>
</tr>
<tr>
<td>35H</td>
<td>IFM code</td>
<td>EX “IFM0Q2-0101”</td>
<td>11</td>
</tr>
<tr>
<td>36H</td>
<td>7816 (CPU) Software revision</td>
<td>EX “V20A9211”</td>
<td>8</td>
</tr>
<tr>
<td>41H</td>
<td>Memory card controller Software revision</td>
<td>EX “V20A9211”</td>
<td>8</td>
</tr>
<tr>
<td>42H</td>
<td>RF card controller Software revision</td>
<td>EX “V20A9211”</td>
<td>8</td>
</tr>
</tbody>
</table>

PM (31H/32H/36H/37H/41H/42H), Data indicates revision and date EX) “V20A9211”, indicates the ICRW is version 2.0 of CRT-350.
8.11 Retract counter command

8.11.1 Retract counter read

Command

```
"C" 43H 30H
```

Positive response

```
"P" 43H 30H ST1 ST0 CT10 CT1
```

Negative response

```
"N" 43H 30H E1 E0
```

This command is only applicable when ICRW is initialized with PM 34H-36H of initialize command, which indicates the current counter value. CT10 and CT1 denotes the retract counter value, and encoded by ASCII, then max is 99. 
Ex) counter value ="90" -> CT10=39H, CT1=30H
8.11.2 Retract counter set

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>43H</th>
<th>31H</th>
<th>CT10</th>
<th>CT1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Positive response</th>
<th>&quot;P&quot;</th>
<th>43H</th>
<th>31H</th>
<th>ST1</th>
<th>ST0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Negative response</th>
<th>&quot;N&quot;</th>
<th>43H</th>
<th>31H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
</table>

This command is only applicable when ICRW is initialized with PM 34H-36H of initialize command, which indicates the current counter value.
Acceptable value range is between 0 and 99, indicated by CT10 and CT1, and encoded by ASCII
Ex) in case of setting “90” in retract counter.->CT10=39H, CT1=30H
8.12 CPU card control command

8.12.1 Activate CPU card command (cold reset)

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>49H</th>
<th>30H</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>49H</td>
<td>30H</td>
<td>ST1</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>49H</td>
<td>30H</td>
<td>E1</td>
</tr>
</tbody>
</table>

To cold reset IC card, ICRW supplies power (VCC) and clock (CLK), then reset (RST) release.

VC=30H: ICRW supplies with +5V to VCC and activates in line with the EMV2000 Ver 4.0
VC=33H: ICRW supplies with +5V to VCC and activates in line with the ISO/IEC7816-3
VC=35H: ICRW supplies with +3V to VCC and activates in line with the ISO/IEC7816-3: 1997

After ATR reception, ICRW supplies voltage to VCC in accordance with the value of ART on T=15

VC=36H: ICRW supplies with +5V to VCC and activates in line with the ISO/IEC7816-3: 1997/Amd 1:2001

After ATR reception, ICRW supplies voltage to VCC in accordance with the value of ART on T=15

VC=40H: ICRW supplies with +5V to VCC and activates in line with the MONEO

In case there is no VC word, it will have 30H as default value.

Also, Answer To Reset (ATR) from ICC is received and transmitted to HOST.

| TS | TO | TA1 | TB1 | TCK |

When a power failure is recognized while a power supply is supplied to the card, error code "60" is returned.

If ATR receive error is occurred, error code “61”“63”“64” are returned.

When content of ATR is not based on such protocol which is supported by EMV2000 ver4.0, ICRW sends back error code “69” (This error code is returned when VC=30H).
When content of ATR is not based on such protocol which is supported by ISO/IEC7816-3, ICRW sends back error code “66” (This error code is returned when VC=33H/35H/36H).
### 8.12.2 Deactivate CPU card command

<table>
<thead>
<tr>
<th>Command</th>
<th>Hex 1</th>
<th>Hex 2</th>
<th>Hex 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>49H</td>
<td>31H</td>
<td></td>
</tr>
</tbody>
</table>

**Positive response**

<table>
<thead>
<tr>
<th>Command</th>
<th>Hex 1</th>
<th>Hex 2</th>
<th>Hex 3</th>
<th>Hex 4</th>
<th>Hex 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td>49H</td>
<td>31H</td>
<td>ST1</td>
<td>ST0</td>
<td></td>
</tr>
</tbody>
</table>

**Negative response**

<table>
<thead>
<tr>
<th>Command</th>
<th>Hex 1</th>
<th>Hex 2</th>
<th>Hex 3</th>
<th>Hex 4</th>
<th>Hex 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td>49H</td>
<td>31H</td>
<td>E1</td>
<td>E0</td>
<td></td>
</tr>
</tbody>
</table>

This deactivates IC card.
8.12.3 Inquire CPU card Status and working frequency command

**Command**

```
"C" 49H 32H
```

**Positive response**

```
"P" 49H 32H ST1 ST0 STI
```

**Negative response**

```
"N" 49H 32H E1 E0
```

CRW tells the status of CPU card with STI (1 byte)

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

- **b0**: = 1 Active, = 0 Inactive
- **b1**: = 1 +3V, = 0 +5V
- **b3/b2**: = 00 No clock, = 01 3.58 MHz, = 10 7.17 MHz
- **b7 - b4** = ’0100’

Only as b0 = 1, then b1/b2/b3 is valid.

While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card). An error “60” is returned when a power failure is detected.
8.12.4 C-APDU (T=0)

Command

| "C" | 49H | 33H | C-APDU |

Positive response

| "P" | 49H | PX | ST1 | ST0 | R-APDU |

Negative response

| "N" | 49H | 33H | E1 | E0 |

This exchanges data between CPU card and HOST by protocol T=0.

APDU: Application protocol data unit
C-APDU: Command APDU
R-APDU: Response APDU

C-APDU

| CLA | INS | P1 | P2 | LC | Data1 | ...... | Data(N) | LE |

The length range of C-APDU is 4byte to 262 byte, If HOST sends more than 262bytes, error code “04” is sent

R-APDU

| Data1 | ...... | Data(N) | SW1 | SW2 |

PX =33H : The CPU card’s data is 1000 bytes or less
PX =35H : The IC card’s data is 1001 bytes or more
ICRW requires transmitting next IC card’s data, HOST needs to receive the remaining data by using "CI7" command.
An error “60” is returned when a power failure is detected;
If protocol type of IC card is not T=0, error code “62” is sent;
If ICC does not respond within WWT (Working Wait Time), ICRW deactivates an IC card and error code “63” is sent;
If any other protocol error occurs, ICRW deactivates an IC card and error code "64" is sent

Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of CPU card is the standard for C-APDU operation.
8.12.5 C-APDU (T=1)

This exchanges data between CPU card and HOST by protocol T=1.
Maximum size of ICRW can handle is 1000 bytes.
Maximum length of R-APDU is 1000 bytes.
When data for transmitting is 1001 bytes or more, HOST should use “CI5” and "CI6”

PX =33H : The CPU card’s data for R-APDU is 1000 bytes or less
PX =35H : The CPU card’s data for R-APDU is 1001 bytes or more
ICRW requires transmitting next CPU card’s data
HOST needs to receive the remaining data by using "CI7” command.
PX =3FH : ICRW received the S(ABORT response) block from the CPU card, so suspended transmission, and deactivated the CPU card. HOST should stop following data send.
ICRW executes C-APDU command by T=1 in accordance with protocol, and combines C-APDU sent by HOST into a I-block ,and sends to CPU card, and next step, extracts R-APDU in I-block responded from CPU card to HOST. Flow diagram is shown below:

A. Send C-APDU (. ICRW adds Prologue field and Epilogue field to I-block, and sends to a CPU card)

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>49H</th>
<th>34H</th>
<th>C-APDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>49H</td>
<td>PX</td>
<td>ST1</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>49H</td>
<td>34H</td>
<td>E1</td>
</tr>
</tbody>
</table>
B. ICRW returns “R-APDU” data to HOST (returns R-APDU in I-block from CPU card)

<table>
<thead>
<tr>
<th>Prologue field</th>
<th>Information field</th>
<th>Epilogue field</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAD PCB LE N</td>
<td>CLA INS P1 P2 Lc Data1 ...... Data(Lc) Le</td>
<td>EDC</td>
</tr>
</tbody>
</table>

C-APDU

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

E1, E0 =“02” No card inside ICRW
E1, E0 =“04” HOST sends 1001 bytes or more
E1, E0 =“60” While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
E1, E0 =“62” protocol type of CPU card is not T=1,
E1, E0 =“63” CPU card does not respond within BWT(Block Waiting Time) or CWT(Character Waiting Time)
E1, E0 =“64” data exchange error occurs between ICRW and HOST
E1, E0 =“65” HOST tries to communicate before an CPU card activation,

Please refer to specifications of E1,E0 in list 2
Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of CPU card is the standard for C-APDU operation.
8.12.6 CPU send link block data as T=1 (“C15”)

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>49H</th>
<th>35H</th>
<th>C-APDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Positive response

| "P" | 49H | PX | ST1 | ST0 |

Negative response

| "N" | 49H | 35H | E1 | E0 |

This is used to startup sending C-APDU between HOST and CPU card as T=1.
PXB=37H: ICRW requires to receive next IC card's data.
HOST needs transmit the remaining data by using command "CI5" or "CI6".
PXB=3FH: ICRW received the S (ABORT response) block from the CPU card, so suspended transmission, and deactivated the CPU card. HOST should stop following data send.

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST.
In case of bad for C-APDU, ICRW sends negative response and error code to HOST.

E1, E0 = “02” No card inside ICRW
E1, E0 = “04” HOST sends 1001 bytes or more
E1, E0 = “06” While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
E1, E0 = “62” protocol type of CPU card is not T=1,
E1, E0 = “63” CPU card does not respond within BWT(Block Waiting Time) or CWT(Character Waiting Time)
E1, E0 = “64” data exchange error occurs between ICRW and HOST
E1, E0 = “65” HOST tries to communicate before an CPU card activation,

Please refer to specifications of E1, E0 in list 2.
Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of CPU card is the standard for C-APDU operation.
8.12.7 CPU send the last link block data as T=1 (“C16”)

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>49H</th>
<th>35H</th>
<th>C-APDU</th>
</tr>
</thead>
</table>

Positive response

| "P" | 49H | PX | ST1 | ST0 |

Negative response

| "N" | 49H | 35H | E1   | E0   |

This command is used when the last data are transmitted.

- PX=34H: The R-APDU's data is 1000 bytes or less.
- PX=35H: The R-APDU's data is 1001 bytes or more.

ICRW requires transmitting next CPU's data.
HOST needs to receive the remaining data by using "CI7" command.

- PX=3FH: ICRW received the S(ABORT response) block from the IC card, so suspended transmission, and deactivated the IC card. HOST should stop following data send.

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

- E1, E0 ="02" No card inside ICRW
- E1, E0 ="04" HOST sends 1001 bytes or more
- E1, E0 ="60" While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
- E1, E0 ="62" protocol type of CPU card is not T=1,
- E1, E0 ="63" CPU card does not respond within BWT (Block Waiting Time) or CWT(Character Waiting Time)
- E1, E0 ="64" data exchange error occurs between ICRW and HOST
- E1, E0 ="65" HOST tries to communicate before an CPU card activation,

Please refer to specifications of E1, E0 in list 2

Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of CPU card is the standard for C-APDU operation.
8.12.8 CPU receive the link block data as T=1(“C17”)

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>49H</th>
<th>35H</th>
<th>C-APDU</th>
</tr>
</thead>
</table>

Positive response

| "P" | 49H | PX | ST1 | ST0 |

Negative response

| "N" | 49H | 35H | E1 | E0 |

This command is used when receiving multi-block data from CPU linking operation.

PX=35H: The CPU card's data is 1001 bytes or more.
ICRW requires transmitting next CPU card's data.
HOST needs to receive the remaining data by using "C17" command.
PX=3FH: ICRW received the S(ABORT response) block from the IC card, so suspended transmission, and deactivated the IC card. HOST should stop following data send.

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

E1, E0 =“02” No card inside ICRW
E1, E0 =“04” HOST sends 1001 bytes or more
E1, E0 =“60” While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
E1, E0 =“62” protocol type of CPU card is not T=1,
E1, E0 =“63” CPU card does not respond within BWT (Block Waiting Time) or CWT(Character Waiting Time)
E1, E0 =“64” data exchange error occurs between ICRW and HOST
E1, E0 =“65” HOST tries to communicate before an CPU card activation,

Please refer to specifications of E1, E0 in list 2
8.12.9 CPU warm reset

Command

"C" 49H 38H

Positive response

"P" 49H 38H ST1 ST0 ATR

Negative response

"N" 49H 38H E1 E0 ATR

ICRW sends a reset pulse, keeping the status of the CPU activated, then returns Response upon receiving "ATR" again (Warm Reset).

This command will take as error when ATR content is not based on such protocol that is supported by this device, ATR from CPU and error code “66” or “69” is sent
8.12.10 CPU card automatic communication (C-APDU command)

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>49H</th>
<th>39H</th>
<th>C-APDU</th>
</tr>
</thead>
</table>

| Positive response | "P" | 49H | PX  | ST1 | ST0 | R-APDU |

| Negative response | "N" | 49H | 39H | E1  | E0  |        |

This exchanges data between IC card by protocol T=0 or T=1. Protocol is recognized automatically, Set Data to "C-APDU".

- **T=0:** Maximum size of data (C-APDU) is 261 bytes
- **T=1:** Maximum size of data (C-APDU) is 1000 bytes

PX=33H: The CPU card's data is 1000 bytes or less.
PX=35H: The CPU card's data is 1001 bytes or more.
ICRW requires transmitting next CPU card's data.
HOST needs to receive the remaining data by using "CI7" command.
PX=3FH: ICRW received the S(ABORT response) block from the IC card, so suspended transmission, and deactivated the IC card. HOST should stop following data send.

While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card).
An error “60” is returned when a power failure is detected.
If protocol type of IC card is not T=0, error code “62” is sent.
If ICC does not respond within appointed time, ICRW deactivates an IC card and error code “63” is sent.
If any other protocol error occurs, ICRW deactivates an IC card and error code "64" is sent.
If HOST tries to communicate before IC card activation, error code “65” is sent.
8.13 SAM (Secure Application Module) control command

8.13.1 Activate SAM command (cold reset)

Command: "C" 49H 40H VC

Positive response: "P" 49H 40H ST1 ST0 ATR

Negative response: "N" 49H 40H E1 E0 ATR

To cold reset SAM. The ICRW supplies power (VCC) and clock (CLK), then reset (RST) release.

VC=30H: ICRW supplies with +5V to VCC and activates in line with the EMV2000 ver4.0.
VC=33H: ICRW supplies with +5V to VCC and activates in line with the ISO/IEC7816-3.

After ATR reception, ICRW supplies voltage to VCC in accordance with the value of ATR on T=15.


After ATR reception, ICRW supplies voltage to VCC in accordance with the value of ATR on T=15.

VC=40H: ICRW supplies with +5V to VCC and activates in line with the MONEO

In case there is no VC word, it will have 30H as default value.

Answer To Reset (ATR) from SAM is received and transmitted to HOST.

TS TO TA1 : TB1 : TCK

When a power failure is recognized while a power supply is supplied to the card, error code "60" is returned.

If ATR receive error is occurred, ICRW initiate the deactivation sequence, and sends back error code “61”, “63”“64”

When content of ATR is not based on such protocol which is supported by EMV2000 ver4.0, ICRW initiate the deactivation sequence, and sends back error code “69”
When content of ATR is not based on such protocol which is supported by ICRW, ICRW initiate the deactivation sequence, and sends back error code “66” (This error code is returned when VC=33H, 35H or 36H).
8.13.2 Deactivate SAM command

<table>
<thead>
<tr>
<th>Command</th>
<th>Positive response</th>
<th>Negative response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>&quot;P&quot;</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>49H 41H</td>
<td>49H 41H ST1 ST0</td>
<td>49H 41H E1 E0</td>
</tr>
</tbody>
</table>

This deactivates SAM.
8.13.3 Inquire SAM Status and working frequency command

Command

```
"C" 49H 42H
```

Positive response

```
"P" 49H 42H ST1 ST0 ST1 STJ
```

Negative response

```
"N" 49H 42H E1 E0
```

This is to require whether SAM activates or not, in case of bad result, then error code ="60"

```
b7 b0
ST1= 0 1 0 0 X X X X (Binary)
```

VCC 1: Active, 0: Inactive

VCC 1: +3V , 0: +5V

00: No clock
01: 3.58 MHz
10: 7.16 MHz
ICRW tells the address of the selected SAM number with STJ

\[
\text{STJ} = \begin{array}{cccccccc}
0 & 0 & 1 & 1 & 0 & X & X & X \\
\end{array} \quad \text{(Binary)}
\]

Selected number
000: SAM 1
001: SAM 2
010: SAM 3
011: SAM 4
100: SAM 5
8.13.4 SAM communication T=0

Command  "C"  49H  43H  C-APDU

Positive response  "P"  49H  PX  ST1  ST0  R-APDU

Negative response  "N"  49H  43H  E1  E0

This exchanges data between SAM by protocol T=0. Set data to "C-APDU".

C-APDU  
CLA  INS  P1  P2  LC  Data1  ……  Data(N)  LE

Maximum size of data ICRW can handle is 261 bytes. If HOST sends 262 bytes or more, error code “04” is sent.

R-APDU  
Data1  ……  Data(N)  SW1  SW2

PX=33H: The SAM's data is 1000 bytes or less.
PX=35H: The SAM's data is 1001 bytes or more.
ICRW requires transmitting next SAM's data. HOST needs to receive the remaining data by using "CIG" command.

While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card). An error “60” is returned when a power failure is detected.
If protocol type of SAM is not T=0, error code “62” is sent.
If SAM does not respond within WWT (Working Wait Time), ICRW deactivates a SAM and error code “63” is sent.
If any other protocol error occurs, ICRW deactivates an SAM and error code "64" is sent.
If HOST tries to communicate before SAM activation, error code “65” is sent.

Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of SAM card is the standard for C-APDU operation.
8.13.5 SAM communication T=1

This exchanges data between SAM by protocol T=1.

Maximum size of ICRW can handle is 1000 bytes. Maximum length of R-APDU is 1000 bytes. When data for transmitting is 1001 bytes or more, HOST should use “Cl5” and “Cl6”

PX=33H: The SAM’s data is 1000 bytes or less
PX=35H: The SAM’s data is 1001 bytes or more
ICRW requires transmitting next SAM’s data
HOST needs to receive the remaining data by using "C17" command.
PX=3FH: ICRW received the S(ABORT response) block from the SAM, so suspended transmission, and deactivated the SAM. HOST should stop following data send.

ICRW executes C-APDU command by T=1 in accordance with protocol, and combines C-APDU sent by HOST into a I-block ,and sends to SAM card, and next step, extracts R-APDU in I-block responded from SAM card to HOST. Flow diagram is shown below:

A. Send C-APDU ( ICRW adds Prologue field and Epilogue field to I-block, and sends to a SAM card)
B.ICRW returns “R-APDU” data to HOST (returns R-APDU in I-block from SAM card)

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

E1, E0 =“02” No card inside ICRW
E1, E0 =“04” HOST sends 1001 bytes or more
E1, E0 =“60” While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
E1, E0 =“62” protocol type of SAM card is not T=1,
E1, E0 =“63” SAM card does not respond within BWT(Block Waiting Time) or CWT(Charater Waiting Time)
E1, E0 =“64” data exchange error occurs between ICRW and HOST
E1, E0 =“65” HOST tries to communicate before an SAM card activation,

Please refer to specifications of E1, E0 in list 2
Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of SAM card is the standard for C-APDU operation.
8.13.6 SAM send link block data as T=1(“CIE”) 

**Command**

```
| Command | C   | 49H | 45H | C-APDU |
```

**Positive response**

```
| Command | P   | 49H | PX  | ST1  | ST0  |
```

**Negative response**

```
| Command | N   | 49H | 45H | E1   | E0   |
```

This is used to startup sending C-APDU between HOST and SAM card as T=1
PX=47H : ICRW requires to receive next IC card's data.
HOST needs transmit the remaining data by using command "CI5" or "CI6".
PX=4FH : ICRW received the S (ABORT response) block from the SAM card, so suspended transmission, and deactivated the SAM card. HOST should stop following data send.

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

- E1, E0 =“02” No card inside ICRW
- E1, E0 =“04” HOST sends 1001 bytes or more
- E1, E0 =“60” While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
- E1, E0 =“62” protocol type of SAM card is not T=1,
- E1, E0 =“63” SAM card does not respond within BWT(Block Waiting Time) or CWT(Character Waiting Time)
- E1, E0 =“64” data exchange error occurs between ICRW and HOST
- E1, E0 =“65” HOST tries to communicate before an SAM card activation,

Please refer to specifications of E1, E0 in list 2
Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of SAM card is the standard for C-APDU operation.
8.13.7 SAM send the last link block data as T=1 (“CIF”)  

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>49H</th>
<th>46H</th>
<th>C-APDU</th>
</tr>
</thead>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;P&quot;</th>
<th>49H</th>
<th>PX</th>
<th>ST1</th>
<th>ST0</th>
<th>R-APDU</th>
</tr>
</thead>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;N&quot;</th>
<th>49H</th>
<th>46H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
</table>

This command is used when the last data are transmitted.
PX=44H: The R-APDU’s data is 1000 bytes or less.
PX=45H: The R-APDU’s data is 1001 bytes or more.
ICRW requires transmitting next SAM's data.
HOST needs to receive the remaining data by using "CI7" command.
PX=4FH: ICRW received the S(ABORT response) block from the IC card, so suspended transmission, and deactivated the IC card. HOST should stop following data send.

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

E1, E0 = “02” No card inside ICRW
E1, E0 = “04” HOST sends 1001 bytes or more
E1, E0 = “06” While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
E1, E0 = “02” protocol type of SAM card is not T=1,
E1, E0 = “03” SAM card does not respond within BWT (Block Waiting Time) or CWT (Character Waiting Time)
E1, E0 = “04” data exchange error occurs between ICRW and HOST
E1, E0 = “05” HOST tries to communicate before an SAM card activation,

Please refer to specifications of E1, E0 in list 2
Note) Please refer to specifications of ISO/IEC7816-3 about length of T=0 APDU, and the COS commands of SAM card is the standard for C-APDU operation.
8.13.8 SAM receive the link block data as T=1(“CIG”) 

<table>
<thead>
<tr>
<th>Command</th>
<th>Positive response</th>
<th>Negative response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot; 49H 47H</td>
<td>&quot;P&quot; 49H PX ST1 ST0 R-APDU</td>
<td>&quot;N&quot; 49H 47H E1 E0</td>
</tr>
</tbody>
</table>

This command is used when receiving multi-block data from SAM linking operation.

PX=45H : The SAM card's data is 1001 bytes or more.
   ICRW requires transmitting next SAM card's data.
   HOST needs to receive the remaining data by using "CI7" command.
PX=4FH : ICRW received the S(ABORT response) block from the IC card, so suspended transmission,
   and deactivated the IC card. HOST should stop following data send.

In case of good for C-APDU, ICRW sends positive response and R-APUD to HOST
In case of bad for C-APDU, ICRW sends negative response and error code to HOST

E1, E0 =“02” No card inside ICRW
E1, E0 =“04” HOST sends 1001 bytes or more
E1, E0 =“60” While a power supply is supplied to the card, the ICRW monitors VCC (the
   power supply line of the card) and a power failure is detected.
E1, E0 =“62” protocol type of SAM card is not T=1,
E1, E0 =“63” SAM card does not respond within BWT (Block Waiting Time) or
   CWT(Character Waiting Time)
E1, E0 =“64” data exchange error occurs between ICRW and HOST
E1, E0 =“65” HOST tries to communicate before an SAM card activation,

Please refer to specifications of E1, E0 in list 2
8.13.9 SAM warm reset

Command

"C"  49H  48H

Positive response

"P"  49H  48H  ST1  ST0  ATR

Negative response

"N"  49H  48H  E1  E0  ATR

ICRW sends a reset pulse, keeping the status of the SAM activated, then returns Response upon receiving "ATR" again (Warm Reset).

This command will take as error when ATR content is not based on such protocol that is supported by this device, ATR from SAM and error code "66" or "69" is sent
8.13.10 SAM card automatic communication (C-APDU command)

This exchanges data between IC card by protocol T=0 or T=1. Protocol is recognized automatically, Set Data to "C-APDU".

T=0: Maximum size of data (C-APDU) is 261 bytes
T=1: Maximum size of data (C-APDU) is 1000 bytes

PX=43H: The SAM card's data is 1000 bytes or less.
PX=45H: The SAM card's data is 1001 bytes or more.

ICRW requires transmitting next SAM card's data.
HOST needs to receive the remaining data by using "CI7" command.

While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card).
An error “60” is returned when a power failure is detected
If protocol type of IC card is not T=0, error code “62” is sent
If ICC does not respond within appointed time, ICRW deactivates an IC card and error code “63” is sent.
If any other protocol error occurs, ICRW deactivates an IC card and error code "64" is sent.
If HOST tries to communicate before IC card activation, error code “65” is sent.
8.13.11 Select SAM

Command

```
"C" 49H 50H SEL
```

Positive response

```
"P" 49H PX ST1 ST0
```

Negative response

```
"N" 49H 50H E1 E0
```

HOST can select SAM 1- SAM 8

- SEL=30H: SAM1
- SEL=31H: SAM2
- SEL=32H: SAM3
- SEL=33H: SAM4
- SEL=34H: SAM5
- SEL=35H: SAM6
- SEL=36H: SAM7
- SEL=37H: SAM8

SAM command is effective only in the module selection. When Initialize command is executed, SAM 1 will be selected.
8.14 Monitoring for removal command

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>54H</th>
<th>30H</th>
<th>TM10</th>
<th>TM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>54H</td>
<td>30H</td>
<td>ST1</td>
<td>ST0</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>54H</td>
<td>30H</td>
<td>E1</td>
<td>E0</td>
</tr>
</tbody>
</table>

ICRW monitors the removal of the ejected card for a specified time. If the card is removed from the gate during this time, ICRW sends a positive response to HOST. If the card is not removed during this time, a negative response with error code “46” is sent to HOST.

TM10 and TM1 indicate the time value in the form of ASCII code (unit: sec.)
Ex) In case of setting 15 to timer
   TM10=31H   TM1=35H
8.15 On/Off for red and green lights control command

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>80H</th>
<th>PM</th>
<th>LED</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;P&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;N&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>80H</th>
<th>PM</th>
<th>ST1</th>
<th>ST0</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>80H</th>
<th>PM</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is to control the lights (red and green) in entry to on or off

- PM=31H: Red light control
- PM=32H: Green light control

- LED=30H: On
- LED=31H: Off
- LED=32H: Glitter (cycle is 2 sec., on for 1 sec., off for 1 sec.)
8.16 Anti-fishing control (just for E shutter)

8.16.1 Anti-fishing status

<table>
<thead>
<tr>
<th>Command</th>
<th>“C” 63H 38H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>“C” 63H 38H st1 st0 Sts</td>
</tr>
<tr>
<td>Negative response</td>
<td>“N” 63H 38H e1 e0</td>
</tr>
</tbody>
</table>

This command returns: Anti-fishing pin working mode and Anti-fishing sensor status.

<table>
<thead>
<tr>
<th>Sts: b7 b0</th>
<th>0 0 1 1 0 0 PIN UP</th>
</tr>
</thead>
</table>

PIN: Anti-fishing pin working mode
- 0: Anti-fishing pin is not activated
- 1: Anti-fishing pin is activated (Automatically or by control command “Cc:” )

UP: Anti-fishing pin sensor status
- 0: Anti-fishing pin is not up
- 1: Anti-fishing pin is up

<table>
<thead>
<tr>
<th>PIN UP Status</th>
<th>PIN UP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>Anti-fishing pin is not activated, Pin is not up (Irregular)</td>
</tr>
<tr>
<td>0 1</td>
<td>Anti-fishing pin is not activated, pin is up (correctly)</td>
</tr>
<tr>
<td>1 0</td>
<td>Anti-fishing pin is activated, pin is not up (correctly)</td>
</tr>
<tr>
<td>1 1</td>
<td>Anti-fishing pin is activated, pin is up (Irregular)</td>
</tr>
</tbody>
</table>
8.16.2 Anti-fishing mode setting

<table>
<thead>
<tr>
<th>Command</th>
<th>“C” 63H 39H Enb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>“C” 63H 39H St1 St0</td>
</tr>
<tr>
<td>Negative</td>
<td>“N” 63H 39H e1 E0</td>
</tr>
</tbody>
</table>

This command returns: Anti-fishing pin working mode

- **Enb**: Anti-fishing mode
  - **30H**: Disable (Default)
    - Anti-fishing pin will not be up automatically when error happened (See error code)
  - **31H**: Enable
    - In this mode, the pin is up automatically when error happened (See error code), to avoid card being drawn out from the gate direction.
    - The pin that is already up automatically can be released by command “Cc:1”
    - Even the pin is in enable mode, if there is no card inside ICRW, the pin will not operate.

This is EEPROM command, Enable mode continues until it sets it to Disable mode or power turn off.

**Error code:**
- Error “10”: Card jam
- Error “11”: Shutter failure (open/close forcibly)
8.16.3 Anti-fishing pin control command

<table>
<thead>
<tr>
<th>Command</th>
<th>“C” 63H 3AH move</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>“C” 63H 3AH st1 st0</td>
</tr>
<tr>
<td>Negative response</td>
<td>“N” 63H 3AH e1 e0</td>
</tr>
</tbody>
</table>

This is to control the anti-fishing pin.
This command can be executed even if the anti-fishing mode has been enabled.

Move: To control Anti-fishing pin

30H: The pin is up
Pin is up successfully, ICRW returns positive response.
Pin fails to be up, ICRW returns error “54”.
When receiving the command after execution of this command, ICRW returns error “55”.

31H: Release Anti-fishing pin
Anti-fishing pin release success, ICRW returns positive response
Anti-fishing pin release failure, ICRW returns error “54”.

In Enable mode of anti-fishing, if the pin is up because card jam or error happens, Host must execute
This command is to release the pin.
After executing this command, ICRW needs initialize command to clear error(Card jam/shutter error)
8.16.4 Anti-fishing unit setting

Command

```
“C” 63H 3FH cmd
```

Positive response

```
“C” 63H 3FH st1 st0
```

Negative response

```
“N” 63H 3FH e1 e0
```

This command is to set the presence of anti-fishing unit.
The condition of setup is indicated in the response of initialize command.

cmd: Set up the anti-fishing unit

- 30H: absent
- 31H: present

Note:
1) This setting is EEPROM command, it does not change by Initialize command or power on/off.
2) This setting is necessary when the unit is assembled or it is removed.
8.17 Automatically Detect Card Type

8.17.1 Automatically detect IC card type

Command

<table>
<thead>
<tr>
<th>C</th>
<th>90H</th>
<th>30H</th>
</tr>
</thead>
</table>

Positive response

<table>
<thead>
<tr>
<th>P</th>
<th>90H</th>
<th>30H</th>
<th>ST1</th>
<th>ST0</th>
<th>IC_card_type</th>
</tr>
</thead>
</table>

Negative response

<table>
<thead>
<tr>
<th>N</th>
<th>90H</th>
<th>30H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
</table>

After receiving this command, ICRW moves card to IC card operation position, and then resets IC card to get ATR which will be selected to detect IC card type. This is used in Comprehensive treatment in IC card.

<table>
<thead>
<tr>
<th>IC_card_type (2 byte)</th>
<th>Card type</th>
</tr>
</thead>
<tbody>
<tr>
<td>“0”</td>
<td>“0”</td>
</tr>
<tr>
<td>“1”</td>
<td>“0”</td>
</tr>
<tr>
<td>“1”</td>
<td>“1”</td>
</tr>
<tr>
<td>“2”</td>
<td>“1”</td>
</tr>
<tr>
<td>“2”</td>
<td>“2”</td>
</tr>
<tr>
<td>“3”</td>
<td>“3”</td>
</tr>
<tr>
<td>“4”</td>
<td>“4”</td>
</tr>
<tr>
<td>“5”</td>
<td>“5”</td>
</tr>
<tr>
<td>“6”</td>
<td>“6”</td>
</tr>
<tr>
<td>“7”</td>
<td>“7”</td>
</tr>
<tr>
<td>“8”</td>
<td>“8”</td>
</tr>
<tr>
<td>“9”</td>
<td>“9”</td>
</tr>
<tr>
<td>“3”</td>
<td>“1”</td>
</tr>
<tr>
<td>“4”</td>
<td>“1”</td>
</tr>
<tr>
<td>“5”</td>
<td>“1”</td>
</tr>
<tr>
<td>“5”</td>
<td>“1”</td>
</tr>
</tbody>
</table>

In the case of good detecting, ICRW sends positive response and IC_card_type to HOST.

In the case of bad detecting, ICRW sends negative response and error code to HOST.
E1, E0=“02” No card inside ICRW, command can not be executed
E1, E0= “60” a power failure is detected

Note) After detecting card type, ICRW deactivates IC card, so it needs to reset relative type IC card before its operation.
As case of unknown card type in detecting, ICRW retrieves IC card to relative operation position to detect card type for one time
8.17.2 Automatically detect RF card type

Command

"C" 90H 32H

Positive response

"P" 90H 32H ST1 ST0 RF_card_type

Negative response

"N" 90H 32H E1 E0

After receiving this command, ICRW moves card to RF card operation position to detect card type, and returns relative card type.

<table>
<thead>
<tr>
<th>Cart_type(2 byte)</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>'0' '0'</td>
<td>Unknown RF type</td>
</tr>
<tr>
<td>'1' '0'</td>
<td>Mifare one S50 type</td>
</tr>
<tr>
<td>'1' '1'</td>
<td>Mifare one S70 type</td>
</tr>
<tr>
<td>'2' '0'</td>
<td>Mifare one UL type</td>
</tr>
<tr>
<td>'2' '0'</td>
<td>Type A CPU type</td>
</tr>
<tr>
<td>'3' '0'</td>
<td>Type B CPU type</td>
</tr>
</tbody>
</table>

In the case of good detecting, ICRW sends positive response and RF_card_type to HOST.

In the case of bad detecting, ICRW sends negative response and error code to HOST.

E1, E0=“02”  No card inside ICRW, command can not be executed

Note) After detecting card type, ICRW deactivates RF card, so it needs to activate/research RF card before its operation.
8.18 EEPROM read/write(omit)
8.19 Lifetime of entry, MAG. head, IC card device

8.19.1 Gain the Lifetime of entry, MAG. head, IC card device

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>A1H</th>
<th>30H</th>
<th>SH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>A1H</td>
<td>30H</td>
<td>ST1</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>A1H</td>
<td>30H</td>
<td>E1</td>
</tr>
</tbody>
</table>

This is to get the lifetime of Wearing parts.

- SH=30H: Times of strobe in entry for opening
- SH=31H: Times of leaping MAG. head, one time during one leaping
- SH=32H: Lifetimes of IC card device

Notes) 1. Length of R-data is 8 bytes, first byte is the highest order, the last one is the lowest order, every byte is indicated by 30H-39H as the 0-9
   ex) 1, 450, 001, Sn =30H 31H 34H 35H 30H 30H 30H 31H
   2. Every time is recorded in RAM, as for 100 times, then clear RAM to zero, and increase one time in EEPROM, if power down during counter less than 100, then retrieve to counter by clearing RAM.
   3. Reset command is valid for the clearing of RAM.
8.19.2 Initialize the Lifetime of entry, MAG. head, IC card device

Command

"C" A1H 31H SH W-data

Positive response

"P" A1H 31H ST1 ST0

Negative response

"N" A1H 31H E1 E0

This is to initialize the lifetime of Wearing parts. W-data is the same as R-data in 8.19.1.
8.20 Read/write ICRW’s SN

8.20.1 Read ICRW SN

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>A2H</th>
<th>30H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>A2H</td>
<td>30H</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>A2H</td>
<td>30H</td>
</tr>
</tbody>
</table>

This is to get the SN of ICRW. (The serial code for ICRW maintaining)
“SN” is form of 18 alphabets and Arabic numerals
Ex) 2008-12-27, CRT-350-（002）MR-LDN, then “SN” = S350003MR081217001, and external label is S081217001
“S” indicates the serial number is made in production line; “F” indicates ICRW is maintained for first time, “U” indicates ICRW is maintained for second time, “W” indicates ICRW is maintained for third time
The external label will be changed during maintain, but not for “SN”
350: indicates the mode number for ICRW, Ex) CRT-350-（002）MR-LDN->（XXX）
MR: Indicates the main function of ICRW
081217001: indicates it’s the product date, it is the No. 1 ICRW on 2008-12-27
8.20.2 Write ICRW’s SN (omit)
8.21 Read Configured characters

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>ASCII Code</th>
<th>Hex Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td></td>
<td>A3H 30H</td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>ASCII Code</th>
<th>Hex Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td></td>
<td>A3H 30H ST1 ST0 S</td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>ASCII Code</th>
<th>Hex Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td></td>
<td>A3H 30H E1 E0</td>
</tr>
</tbody>
</table>

S1=5,  indicates it is CRT-350
S2/S3/S4: Indicates it is client code and serial number
S5: Indicates card type recognized by ICRW
S6: Indicates the communication interface
S7: Indicates magnetic card type recognized by ICRW
S8: Indicates entry type in ICRW
S9: Indicates whether holding PSIM board or not
Every character is denoted by ASCII code in the instruction of ICRW
Ex):  S=5001MRLDN ->CRT-350-（001）MR-LDN
Annex Ⅰ  Contact IC card control command

1. Memory card reset command

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>91H</th>
<th>30H</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>91H</td>
<td>30H</td>
<td>ST1 ST0 IC_Reset data</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>91H</td>
<td>30H</td>
<td>E1 E0</td>
</tr>
</tbody>
</table>

Reset relative type memory card, and return response data.
VCC: The choice of a power supply voltage to supply
- =30H ICRW supplies with +5V to VCC
- =31H ICRW supplies with +3V to VCC
In the case of no VCC, the default value of VCC is 30H
SC and SC1: indicates reset appointed memory card (as shown below), SC1 is only valid for 24CXX card.

<table>
<thead>
<tr>
<th>SC</th>
<th>SC1</th>
<th>Function (Card reset)</th>
<th>IC_Reset data (when positive response)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30H</td>
<td>Null</td>
<td>SLE4442</td>
<td>“A2 13 10 91”</td>
</tr>
<tr>
<td>31H</td>
<td>Null</td>
<td>SLE4428</td>
<td>“92 23 10 91”</td>
</tr>
<tr>
<td>32H</td>
<td>30H</td>
<td>24C01</td>
<td>“24C01”</td>
</tr>
<tr>
<td></td>
<td>31H</td>
<td>24C02</td>
<td>“24C02”</td>
</tr>
<tr>
<td></td>
<td>32H</td>
<td>24C04</td>
<td>“24C04”</td>
</tr>
<tr>
<td></td>
<td>33H</td>
<td>24C08</td>
<td>“24C08”</td>
</tr>
<tr>
<td></td>
<td>34H</td>
<td>24C16</td>
<td>“24C16”</td>
</tr>
<tr>
<td></td>
<td>35H</td>
<td>24C32</td>
<td>“24C32”</td>
</tr>
<tr>
<td></td>
<td>36H</td>
<td>24C64</td>
<td>“24C64”</td>
</tr>
<tr>
<td></td>
<td>37H</td>
<td>24C128</td>
<td>“24C128”</td>
</tr>
<tr>
<td></td>
<td>38H</td>
<td>24C256</td>
<td>“24C256”</td>
</tr>
<tr>
<td>33H</td>
<td>Null</td>
<td>AT45D041</td>
<td>“AT45D041”</td>
</tr>
<tr>
<td>34H</td>
<td>Null</td>
<td>AT88SC1608</td>
<td>“2C 55 AA A0”</td>
</tr>
</tbody>
</table>
In the case of good reset, ICRW sends positive response and Reset_data to HOST
In the case of bad reset, ICRW sends negative response and error code to HOST

E1, E0= “02” No card inside ICRW
E1, E0= “60” a power failure is detected
E1, E0= “61” Negative reset.

Other error code, please refer to E1 and E0 in list 2
2. Deactivate memory card

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>91H</th>
<th>31H</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>91H</th>
<th>31H</th>
<th>ST1</th>
<th>ST0</th>
<th>SC</th>
<th>ST1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>91H</th>
<th>31H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When this command is received, ICRW deactivates memory card. In the case of good reset, ICRW sends positive response to HOST. In the case of bad reset, ICRW sends negative response to HOST.
3. Inquire Status of memory card

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>SV: current power supply voltage to memory card</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot; 91H 32H</td>
<td>SV =30H the current power supply voltage is zero, and have not reset card</td>
</tr>
<tr>
<td></td>
<td>SV =31H the current power supply voltage is 5V</td>
</tr>
<tr>
<td></td>
<td>SV =32H the current power supply voltage is 3V</td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>P 91H 32H ST1 ST0 IC_card_type STI</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;0&quot; 0 &quot;0&quot; No memory card</td>
</tr>
<tr>
<td>&quot;1&quot; 0 &quot;1&quot; T=0 CPU Card</td>
</tr>
<tr>
<td>&quot;1&quot; 1 &quot;1&quot; T=1 CPU Card</td>
</tr>
<tr>
<td>&quot;2&quot; 0 &quot;2&quot; 24C01 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 2 &quot;3&quot; 24C02 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 3 &quot;4&quot; 24C04 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 4 &quot;5&quot; 24C08 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 5 &quot;6&quot; 24C16 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 6 &quot;7&quot; 24C32 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 7 &quot;8&quot; 24C64 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 8 &quot;9&quot; 24C128 Card</td>
</tr>
<tr>
<td>&quot;2&quot; 9 &quot;A&quot; 24C256 Card</td>
</tr>
<tr>
<td>&quot;3&quot; 1 &quot;B&quot; SLE4442 Card</td>
</tr>
<tr>
<td>&quot;3&quot; 2 &quot;C&quot; SLE4428 Card</td>
</tr>
<tr>
<td>&quot;4&quot; 1 &quot;D&quot; AT45D041 Card</td>
</tr>
<tr>
<td>&quot;5&quot; 1 &quot;E&quot; AT88SC1608 Card</td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>N 91H 32H E1 E0</th>
</tr>
</thead>
</table>

In the case of good reset, ICRW sends positive response and IC_card_type SV to HOST.
In the case of bad reset, ICRW sends negative response and error code to HOST

E1, E0=“02” No card inside ICRW
E1, E0= “60” a power failure is detected

Other error codes, please refer to E1 and E0
3. Memory Card Command

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>91H</th>
<th>33H</th>
<th>T_command_data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>91H</td>
<td>33H</td>
<td>ST1 ST0</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>91H</td>
<td>33H</td>
<td>E1 E0</td>
</tr>
</tbody>
</table>

This includes the operations of read and write and key verification for memory card.

T_Command_data: data packet of memory card sending (shown below table), for details of all operation, please refer to below descriptions

T_Command_data format:

<table>
<thead>
<tr>
<th>CLA</th>
<th>INS</th>
<th>P1</th>
<th>P2</th>
<th>Lc</th>
<th>Data</th>
<th>Le</th>
</tr>
</thead>
</table>

CLA: Type word for the operation of memory card
INS: Control word for the operation of memory card write/read/key verification
P1: High order of start address for the operation of memory card
P2: Low order of start address for the operation of memory card
Lc: Sent data length for the operation of memory card
Data: Command data (data/key in writing card) for the operation of memory card
Le: Expectation returned data length for the operation of memory card

R_Command_data: response data packet for the operation of memory card

R_Command_data format:

<table>
<thead>
<tr>
<th>Data</th>
<th>Sw1</th>
<th>Sw0</th>
</tr>
</thead>
</table>

Data: the returned data for the operation of memory card (with or without)
Sw1: Status word1 for the operation of memory card
Sw0: Status word0 for the operation of memory card

The meaning of Sw1 and Sw0 as shown below
As the case of good read card, ICRW sends positive response and R_command_data to HOST.
Ex)  The last operation result is right just as SW1, SW0 = 9000H in positive response. If SW1, SW0 = 6XXXH, then it is negative operation.

As the case of bad read card, ICRW sends positive response and error code to HOST.

E1, E0 = “02”  No card inside ICRW
E1, E0 = “04”  HOST sends 270 bytes or more T_command_data
E1, E0 = “60”  While a power supply is supplied to the card, the ICRW monitors VCC (the power supply line of the card) and a power failure is detected.
E1, E0 = “62”  Type error in operation of memory card, the sent T_command_data is not included in this protocol
E1, E0 = “65”  it does not reset card, and inhibits executing T_command_data command

Other error codes, please refer to detail instruction in this protocol.
3. 1 T_command_data operation of SLE4442

<table>
<thead>
<tr>
<th>Command</th>
<th>CLA</th>
<th>INS</th>
<th>P1</th>
<th>P2</th>
<th>Lc</th>
<th>Data</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data read from main memory</td>
<td>00H</td>
<td>B0H</td>
<td>00H</td>
<td></td>
<td></td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td>Data read from protection memory</td>
<td>00H</td>
<td>B1H</td>
<td>00H</td>
<td></td>
<td></td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td>Data read from security memory</td>
<td>00H</td>
<td>B2H</td>
<td>00H</td>
<td></td>
<td></td>
<td>Read data length (01H–04H)</td>
<td>null</td>
</tr>
<tr>
<td>Data write to main memory</td>
<td>00H</td>
<td>D0H</td>
<td>00H</td>
<td></td>
<td></td>
<td>Write data</td>
<td>null</td>
</tr>
<tr>
<td>Data write to protection memory</td>
<td>00H</td>
<td>D1H</td>
<td>00H</td>
<td></td>
<td></td>
<td>Write data</td>
<td>null</td>
</tr>
<tr>
<td>Data write to security memory</td>
<td>00H</td>
<td>D2H</td>
<td>00H</td>
<td></td>
<td>03H</td>
<td>PSC data (3 byte)</td>
<td>null</td>
</tr>
<tr>
<td>Verification data</td>
<td>00H</td>
<td>20H</td>
<td>00H</td>
<td>00H</td>
<td>03H</td>
<td>PSC data (3 byte)</td>
<td>null</td>
</tr>
</tbody>
</table>

Write main memory, protection memory, security memory right in the positive key verification.

Note)
As write/read main memory, Lc=00H, and the capacity of the main memory is 256 bytes.
As read protection memory, ICRW can read the protection status in the memory whose range is between 00H and 1FH (“0” indicates write protection, “1” indicates no write protection), all the data of the protection memory are 32bits as on 4bytes, for details, please refer to the read of SLE4442 Card data sheet.
3. 2 T_command_data operation of SLE4428

<table>
<thead>
<tr>
<th>Command</th>
<th>CLA</th>
<th>INS</th>
<th>P1 , P2</th>
<th>Lc</th>
<th>Data</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data read from main memory</td>
<td>01H</td>
<td>B0H</td>
<td>Start address</td>
<td>Null</td>
<td>Null</td>
<td>Read data length (00H–FFH)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0000H–03FFH)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data read from protection bit</td>
<td>01H</td>
<td>B1H</td>
<td>Start address</td>
<td>Null</td>
<td>Null</td>
<td>Read data length (01H–80H)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0000H–03FFH)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data write to main memory</td>
<td>01H</td>
<td>D0H</td>
<td>Start address</td>
<td>Read data length (00H–FFH)</td>
<td>Write data</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0000H–03FFH)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data write to protection bit</td>
<td>01H</td>
<td>D1H</td>
<td>Start address</td>
<td>Read data length (00H–FFH)</td>
<td>Write data</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0000H–03FFH)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data write to security memory</td>
<td>01H</td>
<td>D2H</td>
<td>03FEH</td>
<td>02H</td>
<td>PSC data</td>
<td>Null</td>
</tr>
<tr>
<td>Verification data</td>
<td>01H</td>
<td>20H</td>
<td>03FEH</td>
<td>02H</td>
<td>PSC data</td>
<td>Null</td>
</tr>
</tbody>
</table>

Write main memory, protection bit, security memory right in the positive key verification.

Note) The byte number Lc="00" of data to write means 256bytes in the main memory.

The capacity of the main memory is 1024 bytes. all the data of the protection bit are 1024bits as on 128bytes.(“0” indicates write protection, “1” indicates no write protection).

As for write of main memory, the last three memory (03FDH–03FFH) of SLE4428 are security memory, memory of “03FDH” is the key verification error counter, memory of “03FEH, 03FFH” is the key data, so do not write these memories random.
3.3 T_command_data operation of I2C Memory Card 24CXX Card

<table>
<thead>
<tr>
<th>Command</th>
<th>CLA</th>
<th>INS</th>
<th>P1</th>
<th>P2</th>
<th>Lc</th>
<th>Data</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of page write</td>
<td>02H</td>
<td>A4H</td>
<td>0000H</td>
<td></td>
<td>01H</td>
<td>Page_len</td>
<td>Null</td>
</tr>
<tr>
<td>Data read to main memory</td>
<td>02H</td>
<td>B0H</td>
<td>Start address (0000H~7FFFH)</td>
<td>Null</td>
<td>Null</td>
<td>Read data length (00H~FFH)</td>
<td></td>
</tr>
<tr>
<td>Data write to main memory</td>
<td>02H</td>
<td>D0H</td>
<td>Start address (0000H~7FFFH)</td>
<td>write data length (01H~80H)</td>
<td>Write data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ICRW can read/write these types cards of 24C01, 24C02, 24C04, 24C08, 24C16, 24C32, 24C64, 24C128, 24C256, and note the valid address range during operation.

To promote the write card speed by selecting the page length (Page_len) for these card.

- Page_len=30H 8 bytes (default write mode)
- Page_len=31H 16 byte
- Page_len=32H 32 byte
- Page_len=33H 64 byte

<table>
<thead>
<tr>
<th>Card_type</th>
<th>address</th>
<th>Page_write length</th>
<th>capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>24C01</td>
<td>0000H~007FH</td>
<td>8</td>
<td>128 byte</td>
</tr>
<tr>
<td>24C02</td>
<td>0000H~00FFH</td>
<td>16</td>
<td>256 byte</td>
</tr>
<tr>
<td>24C04</td>
<td>0000H~01FFH</td>
<td>16</td>
<td>512 byte</td>
</tr>
<tr>
<td>24C08</td>
<td>0000H~03FFH</td>
<td>16</td>
<td>1024 byte</td>
</tr>
<tr>
<td>24C16</td>
<td>0000H~07FFH</td>
<td>16</td>
<td>2048 byte</td>
</tr>
<tr>
<td>24C32</td>
<td>0000H~0FFFH</td>
<td>32</td>
<td>4096 byte</td>
</tr>
<tr>
<td>24C64</td>
<td>0000H~1FFFH</td>
<td>32</td>
<td>8192 byte</td>
</tr>
<tr>
<td>24C128</td>
<td>0000H~3FFFH</td>
<td>64</td>
<td>16384 byte</td>
</tr>
<tr>
<td>24C256</td>
<td>0000H~7FFFH</td>
<td>64</td>
<td>32768 byte</td>
</tr>
</tbody>
</table>

For the page length (Page_write length), please refer to the technology (Memory and Data Sheet)
### 3. 4 T_command_data operation of AT45DB041 Card

<table>
<thead>
<tr>
<th>Command</th>
<th>CLA</th>
<th>INS</th>
<th>P1, P2</th>
<th>Lc</th>
<th>Data</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data read to main memory</td>
<td>03H</td>
<td>B0H</td>
<td>Start address (0000H~7FFFH)</td>
<td>Null</td>
<td>Null</td>
<td>Read data length 01H</td>
</tr>
<tr>
<td>Data write to main memory</td>
<td>03H</td>
<td>D0H</td>
<td>Start address (0000H~7FFFH)</td>
<td>write data length 01H</td>
<td>Write data</td>
<td>Null</td>
</tr>
</tbody>
</table>

This card is operated by page mode, the length of a page is 264 bytes.
P1, P2: Page address
Lc /Le: 01H, indicates a page is to be operated
### 3. 5 T_command_data operation of AT88SC1608 Card

<table>
<thead>
<tr>
<th>Command</th>
<th>CLA</th>
<th>INS</th>
<th>P1</th>
<th>P2</th>
<th>Lc</th>
<th>Data</th>
<th>Le</th>
<th>Data</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data read to main memory</td>
<td>04H</td>
<td>B0H</td>
<td>SnH</td>
<td></td>
<td></td>
<td>Null</td>
<td></td>
<td>Null</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuse-read</td>
<td>04H</td>
<td>B1H</td>
<td>00H</td>
<td>00H</td>
<td></td>
<td>Null</td>
<td></td>
<td>Null</td>
<td>03H</td>
</tr>
<tr>
<td>Data write to main memory</td>
<td>04H</td>
<td>D0H</td>
<td>SnH</td>
<td></td>
<td></td>
<td>Write</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuse-write</td>
<td>04H</td>
<td>D1H</td>
<td>00H</td>
<td>00H</td>
<td>00H</td>
<td>Null</td>
<td></td>
<td>Null</td>
<td></td>
</tr>
<tr>
<td>Initialize authentication memory</td>
<td>04H</td>
<td>D2H</td>
<td>00H</td>
<td></td>
<td>08H</td>
<td>Qdata</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key verification</td>
<td>04H</td>
<td>20H</td>
<td>PnH</td>
<td>00H</td>
<td>03H</td>
<td>PSC data</td>
<td>Null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication verification</td>
<td>04H</td>
<td>21H</td>
<td>00H</td>
<td>00H</td>
<td>08H</td>
<td>Qdata</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sn : Area number, 8 application areas (128bytes for one area) and control areas (128bytes for all)

Sn = 30H ~ 37H (application area No.1 ~ application area No.8)
Sn = 38H control area (date write to this area will be in main key verification)

Pn : write/read key verification

Pn = 30H ~ 37H (read key of application area No.1~application area No.8)
Pn = 40H ~ 47H (write key of application area No.1~application area No.8, the main key is the one in application area No.8)

Qdata: initialize authentication memory data: Q0~Q7
Qdata: authentication verification data: Q0~Q7

Note) the returned three bytes data in read fuse:

- **FAE** Fuse: 30H -> fusion, 31H-> no fusion
- **CMA** Fuse: 30H -> fusion, 31H-> no fusion
- **PER** Fuse: 30H -> fusion, 31H-> no fusion

For details , please refer to AT88SC1608 Card Data Sheet.
Annex II Siemens memory card control command

1. Activate Siemens memory card

<table>
<thead>
<tr>
<th>Command</th>
<th>Positive response</th>
<th>Negative response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot; 52H 30H</td>
<td>&quot;P&quot; 52H 30H ST1 ST0 ATR</td>
<td>&quot;N&quot; 52H 30H E1 E0</td>
</tr>
</tbody>
</table>

To activate the memory cards, ICRW supply power (VCC) and clock (CLK), and assert reset (RST) signal. Then, the memory card is activated and returns ATR. ICRW returns a negative response when proper ATR isn't received from the memory card.
2. Deactivate Siemens memory card

Command

"C"  52H  31H

Positive response

"P"  52H  31H  ST1  ST0

Negative response

"N"  52H  31H  E1  E0

After deactivate, card should be re-activated to do other operations
3. Inquire Status of Siemens memory card

Command

"C" 52H 32H

Positive response

"P" 52H 32H ST1 ST0 ST1

Negative response

"N" 52H 32H E1 E0

This command reports the status of Siemens memory card in "STI" byte.

ST1= 0 1 0 0 X X X X (Binary)

- VCC 1: Active, 0: Inactive
- Sort 1: SLE4428, 0: SLE4442
- Not Available
- Not Available
4. Communicate with SLE4442

After the command was executed properly, ICRW returns a positive response with status information in response data "SW1+SW2".

4.1 Data read

4.1.1 Data read from main memory on SLE4442

ICRW reads data from the main memory of SLE4442

RSD= 00B0H: Start word
ADR=0x00− 0xFF: The start address to read data in the main memory
LEN=0x01-0x80: The number of bytes of data to read(range of 1byte to 128bytes)
In the case of good read, all the read data are in “DATA”
Note)
The number of bytes of data in main memory is 256 bytes, pay attention to the valid range of address and length
4.1.2 Data read from protection memory on SLE4442

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>52H</th>
<th>33H</th>
<th>RSD</th>
<th>ADR</th>
<th>LEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Command</th>
<th>52H</th>
<th>33H</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Command</th>
<th>52H</th>
<th>33H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ICRW reads data from the protection memory of SLE4442

RSD= 00B1H: Start word
ADR: The start address to read data in the protection memory
LEN: The number of bytes of data to read

In the case of good read, all the read data are in “DATA”
4.1.3 Data read from PSC on SLE4442

Command

"C" 52H 33H RSD ADR LEN

Positive response

"P" 52H 33H ST1 ST0 DATA

Negative response

"N" 52H 33H E1 E0

ICRW reads data from the PSC of SLE4442

RSD= 00B2H: Start word
ADR: The start address to read data in the PSC
LEN: The number of bytes of data to read

In the case of good read, all the read data are in “DATA”
4.2 Data write

4.2.1 Data write to main memory on SLE4442

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>52H 33H WSD ADR LEN DAT</td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
</tr>
</tbody>
</table>

ICRW writes data in the main memory.

WSD=00D0H: Start word
ADR=0x00~0xFF: the start address to write data in the main memory
LEN=0x01~0x80: the number of bytes of data to write(range of 1byte to 128bytes)
DAT: to be written data
4.2.2 Data write to protection memory on SLE4442

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>52H</th>
<th>33H</th>
<th>WSD</th>
<th>ADR</th>
<th>LEN</th>
<th>DAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>52H</td>
<td>33H</td>
<td>ST1</td>
<td>ST0</td>
<td>DATA</td>
<td></td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>52H</td>
<td>33H</td>
<td>E1</td>
<td>E0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ICRW writes data in the protection memory.

- **WSD=00D1H**: Start word.
- **ADR=0x00~0x31**: The start address to write data in the protection memory.
- **LEN=0x20**: The number of bytes of data to write (32BYTES).
- **DAT**: To be written data.

Note

ICRW can set up writing protection in a part (00H-31H) of the main memory which can be protected. Once it is set up, the protection can't be canceled.
4.3 Verification key to SLE4442

Command

"C" 52H 33H CSD LEN CMP

Positive response

"P" 52H 33H ST1 ST0 DATA

Negative response

"N" 52H 33H E1 E0

CSD=00200000H: Start word
LEN=03H: the number of bytes of data to compare
CMP: the data to compare
4.4 Modify key to SLE4442

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>52H</th>
<th>33H</th>
<th>MSD</th>
<th>LEN</th>
<th>CMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>&quot;P&quot;</td>
<td>52H</td>
<td>33H</td>
<td>ST1</td>
<td>ST0</td>
<td>DATA</td>
</tr>
<tr>
<td>Negative response</td>
<td>&quot;N&quot;</td>
<td>52H</td>
<td>33H</td>
<td>E1</td>
<td>E0</td>
<td></td>
</tr>
</tbody>
</table>

MSD=00240100H: Start word
LEN=03H: Length of new keys
DAT: New keys
### 5 Communicate with SLE4428

After the command was executed properly, ICRW returns a positive response with status information in response data "SW1+SW2".

#### 5.1 Data read

**5.1.1 Data Reading of main-memory of SLE4428**

<table>
<thead>
<tr>
<th>Command</th>
<th>C</th>
<th>52H</th>
<th>34H</th>
<th>RSD</th>
<th>ADR</th>
<th>LEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive response</td>
<td>P</td>
<td>52H</td>
<td>34H</td>
<td>ST1</td>
<td>ST0</td>
<td>DATA</td>
</tr>
<tr>
<td>Negative response</td>
<td>N</td>
<td>52H</td>
<td>34H</td>
<td>E1</td>
<td>E0</td>
<td></td>
</tr>
</tbody>
</table>

RSD= 00B0H: Start word  
ADR=0x000– 0x3FF: The start address to read data in the main memory  
LEN=0x01-0x80: The number of bytes of data to read (range of 1byte to 128bytes)  
In the case of good read, all the read data are in “DATA”  
Note)  
The number of bytes of data in main memory is 1Kbytes, pay attention to the valid range of address and length
5.1.2 Condition data reading of protection-bit of SLE4428

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>52H 34H RSD ADR LEN</td>
</tr>
</tbody>
</table>

Positive response

<table>
<thead>
<tr>
<th>Positive response</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td>52H 34H ST1 ST0 DATA</td>
</tr>
</tbody>
</table>

Negative response

<table>
<thead>
<tr>
<th>Negative response</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td>52H 34H E1 E0</td>
</tr>
</tbody>
</table>

RSD= 00B1H: Start word
ADR=0x000–0x3FF: The start address to read data in protection-bit
LEN=0x01-0x80: The number of bytes of data to read (range of 1 byte to 128 bytes)
In the case of good read, all the read data are in “DATA”

Note

The number of bytes of data in main memory is 1K bytes, pay attention to the valid range of address and length
### 5.2 Data write

#### 5.2.1 Data writing to main-memory of SLE4428

<table>
<thead>
<tr>
<th>Command</th>
<th>“C”</th>
<th>52H</th>
<th>34H</th>
<th>WSD</th>
<th>ADR</th>
<th>LEN</th>
<th>DAT</th>
</tr>
</thead>
</table>

Positive response:

<table>
<thead>
<tr>
<th>Command</th>
<th>“P”</th>
<th>52H</th>
<th>34H</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
</table>

Negative response:

<table>
<thead>
<tr>
<th>Command</th>
<th>“N”</th>
<th>52H</th>
<th>34H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
</table>

WSD=00D0H: Start word  
ADR=0x000~0x3FF: The start address to write data  
LEN=0x01~0x80: The number of bytes of data to read (range of 1byte to 128bytes)  
DAT: To be written data  

Note)  
The last three memories (0x03FD, 0x3FE, 0x03FF) are key error counter, do not write it random.
5. 2.2 Data writing to main-memory of SLE4428 (with protecting it)

Command

```
"C" 52H  34H  WSD  ADR  LEN  DAT
```

Positive response

```
"P" 52H  34H  ST1  ST0  DATA
```

Negative response

```
"N" 52H  34H  E1   E0   DATA
```

WSD=00D1H:  Start word
ADR=0x000~0x3FF:  The start address to write data
LEN=0x01-0x80:  The number of bytes of data to read (range of 1byte to 128bytes)
DAT:  To be written data

Note) The last three memories (0x03FD,0x3FE,0x03FF) are key error counter, do not write it random.
### 5.3 Verification key to SLE4428

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;C&quot;</th>
<th>52H</th>
<th>34H</th>
<th>CSD</th>
<th>LEN</th>
<th>CMP</th>
</tr>
</thead>
</table>

**Positive response**

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;P&quot;</th>
<th>52H</th>
<th>34H</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
</table>

**Negative response**

<table>
<thead>
<tr>
<th>Command</th>
<th>&quot;N&quot;</th>
<th>52H</th>
<th>34H</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
</table>

CSD=00200000H: 
Start word
LEN=02H: 
The number of bytes of data to compare
CMP: 
The data to compare
### 5. 4 Modification key to SLE4428

<table>
<thead>
<tr>
<th>Command</th>
<th>MSD</th>
<th>LEN</th>
<th>DAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C&quot;</td>
<td>52H</td>
<td>34H</td>
<td>MSD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Positive response</th>
<th>MSD</th>
<th>LEN</th>
<th>ST1</th>
<th>ST0</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;P&quot;</td>
<td>52H</td>
<td>34H</td>
<td>ST1</td>
<td>ST0</td>
<td>DATA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Negative response</th>
<th>MSD</th>
<th>LEN</th>
<th>E1</th>
<th>E0</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;N&quot;</td>
<td>52H</td>
<td>34H</td>
<td>E1</td>
<td>E0</td>
</tr>
</tbody>
</table>

MSD=00240H: Start word  
LEN=02H: Length of the new keys  
DAT: New keys
Annex III Contact-less MIFARE ONE card control command

1. Activate/research RF card

Host Command:

```
“C” 93H 30H Set1 Set2
```

(1) Mifare one normal response:

```
“P” 93H 60H 30H St1 St0 Rtype ATQA UID_len UID_data SAK
```

Mifare one error response:

```
“N” 93H 60H 30H e1 e0 Rtype ATQA UID_len UID_data SAK
```

(2) 14443 Type A normal response:

```
“P” 93H 60H 30H St1 St0 Rtype ATQA UID_len UID_data SAK ATS
```

14443 Type A error response:

```
“N” 93H 60H 30H e1 e0 Rtype ATQA UID_len UID_data SAK ATS
```

(3) 14443 Type B normal response:

```
“P” 93H 60H 30H St1 St0 Rtype ATQB
```

14443 Type B error response:

```
“N” 93H 60H 30H e1 e0 Rtype ATQB
```

CRT-350 support reset for IEC/ISO14443 Type A and IEC/ISO 14443 Type B

The reset process is shown as below:

1). Mifare one card:
   1. Request A (REQ A) / Answer Request A (ATQ A).
   2. Anticollision
   3. Select (SEL) / Unique Identifier (UID) & Select Acknowledge (SAK)

Reset successful and return:

ATQA( 2 byte), UID_data (4—10 byte) and SAK( 1 byte).
2).ISO/IEC 14443 Type A:
   1. Request A (REQ A) / Answer Request A (ATQ A).
   2. Anticollision
   3. Select(SEL) / Unique Identifier(UID) & Select Acknowledge(SAK)
   4. Request for answer to select (RATS) / Answer to Select(ATS)
Protocol and parameter selection request(PPSR) / PPS start(PPSS)
Reset successful and return:
ATQA (2 byte), UID_data (4—10 byte), SAK (1 byte), ATS (1-254 byte) and protocol byte (1 byte)

3).ISO/IEC 14443 Type B:
   1. Request B (REQ B) / Answer Request B (ATQ B).
   2. Attribute(ATTRIB) / Answer to ATTRIB

Reset successful and return:
ATQB 12 byte (Including: 50H, PUPI(4 byte), App.data(4 byte), Protocol info(3 byte))

And
Set1, Set2 set operation sequence for different protocol
   41H (‘A’ = Type A), 42H (‘B’ = Type B), 30H (‘0’ = Invalid)

Ex1: Set1 = ‘A’, Set2 = ‘B’ (Default)
Reset sequence: Type A (First), Type B (Second)

Ex2: Set1 = ‘B’, Set2 = ‘A’
Reset sequence: Type B (First), Type A (Second)

Ex3: Set1 = ‘A’, Set2 = ‘0’
Reset sequence: Type A (First), Type B (Do not reset)

Ex4: Set1 = ‘B’, Set2 = ‘0’,
Reset sequence: Type B (First), Type A (Do not reset)

Rtype: Reset protocol
   = 41H (‘A’) ISO/IEC 14443 Type A
   = 42H (‘B’) ISO/IEC 14443 Type B
   = 4DH (‘M’) Philips Mifare one card
<table>
<thead>
<tr>
<th>Communication Protocol</th>
<th>Model No.</th>
<th>CRT-350</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>2010/1/19</td>
<td>3.0</td>
</tr>
<tr>
<td>Ver.</td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td>Page</td>
<td>126/155</td>
<td></td>
</tr>
</tbody>
</table>

Rtype=4DH(‘M’) :
ATQA= 0044H  Mifare Ultralight Card
ATQA= 0004H  Mifare S50 1K Card
ATQA= 0002H  Mifare S70 4K Card

Mafare one, ISO/IEC 14443 Type A return UID_len (The length of UID_data)
- UID_len=4  UID_data is 4 byte
- UID_len=7  UID_data is 7 byte
- UID_len=10 UID_data is 10 byte
2. Deactivate RF card

HOST Command:

```
“C” 93H 60H 31H
```

Normal response:

```
“P” 93H 60H 31H St1 St0
```

Error response:

```
“N” 93H 60H 31H e1 e0
```

Deactivate RF card will close the signal output to antenna
3. Inquire Status of RF card

HOST Command:
```
“C” 93H 60H 32H
```

Normal response:
```
“P” 93H 60H 32H St1 St0 sti stj
```

Error response:
```
“N” 93H 60H 32H e1 e0
```

Current statement sti, stj:

<table>
<thead>
<tr>
<th>sti</th>
<th>stj</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘0’</td>
<td>‘0’</td>
<td>Not Reset</td>
</tr>
<tr>
<td>‘1’</td>
<td>‘0’</td>
<td>Mifare one S50 Card</td>
</tr>
<tr>
<td>‘1’</td>
<td>‘1’</td>
<td>Mifare one S70 Card</td>
</tr>
<tr>
<td>‘2’</td>
<td>‘0’</td>
<td>Mifare one UL Card</td>
</tr>
<tr>
<td>‘2’</td>
<td>‘0’</td>
<td>Type A CPU Card</td>
</tr>
<tr>
<td>‘3’</td>
<td>‘0’</td>
<td>Type B CPU Card</td>
</tr>
</tbody>
</table>
4. MIFARE One Card Control Commands

These functions are specified by a command data form like C-APDU which format is based on T=0 standard.

In this case, CRT-571 recognizes the meaning of the command data, and execute the treatment related to the card by controlling hardware.

After the command was executed properly, CRT-571 returns a positive response with response data 9000H like from the IC card. When an error occurs during the communication with SLE4442, CRT-571 returns a positive response with status information in response data "sw1+sw2" which is based on ISO/IEC 7816-3

<table>
<thead>
<tr>
<th>Sw1</th>
<th>Sw2</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>90H</td>
<td>00H</td>
<td>Success</td>
</tr>
<tr>
<td>6FH</td>
<td>00H</td>
<td>Fail</td>
</tr>
<tr>
<td>6BH</td>
<td>02H</td>
<td>Address overflow</td>
</tr>
<tr>
<td>67H</td>
<td>00H</td>
<td>Address length overflow</td>
</tr>
</tbody>
</table>

4.1 Key verification

Command

```
“C” 93H 60H 33H 00H 20H ks sn lc pdata
```

Positive response

```
“P” 93H 60H 33H St1 St0 rdata
```

Negative response

```
“N” 93H 60H 33H e1 e0
```

Download key to CRT-571 and verify the key directly
ks(1byte): key select (Key A=00H, Key B=01H)
sn(1byte): sector number (S50 card sn=00H-0FH, S70 card sn=00H-27H)
lc(1byte): password length lc=06H
pdata(6 byte): password data
rdata(2 byte): return data
(positive response with data 9000H, and negative response with “sw1+sw2”)

CREATOR
4.2 Verify key from EEPROM

Command

```
"C" 93H 60H 33H 00H 21H ks sn
```

Positive response

```
"P" 93H 60H 33H St1 St0 rdata
```

Negative response

```
"N" 93H 60H 33H e1 e0
```

Read key from EEPROM of RF module and verify the sector key
Download key via command mentioned in 9.10.4.4
EEPROM can preserve 32 groups of key data
ks(1byte): key select                  (Key A=00H, Key B=01H)
sn(1byte): sector number           (sn=00H-0FH)
rdata(2 byte): return data                    (positive response with 9000H)

4.3 Modify sector key (KEY A)

Command

```
"C" 93H 60H 33H 00H D5H 00H sn lc pdata
```

Positive response

```
"P" 93H 60H 33H St1 St0 rdata
```

Negative response

```
"N" 93H 60H 33H e1 e0
```

Modify sector key (key A)
This command only can modify KEY A, and modify KEY B as “0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF” in the mean timemodify control words as “0xFF, 0x07, 0x80, 0x69” (ex-work default)
Use block command to modify Key A, Key B control word
sn(1byte): sector number           (S50 card sn=00H-0FH, S70 card sn=00H-27H)
lc(1byte): password length lc=06H
pdata: password data          6 byte.
rdata(2 byte): return data
(positive response with data 9000H, and negative response with “sw1+sw2”)

4.4 Download password to EEPROM

Command

```
“C”  93H  60H  33H  00H  D0H  ks  sn  lc  pdata
```

Positive response

```
“P”  93H  60H  33H  St1  St0  rdata
```

Negative response

```
“N”  93H  60H  33H  e1  e0
```

Read key from EEPROM of RF module and verify the sector key. EEPROM can preserve 32 groups of key data:

- ks (1 byte): key select (Key A=00H, Key B=01H)
- sn (1 byte): sector number (sn=00H-0FH)
- lc (1 byte): password length lc=06H
- pdata (6 byte): password data
- rdata (2 byte): return data

Positive response: sw1+sw2=9000H
Negative response: sw1+sw2=6F00H
4.5 Read sector data

Command

```
“C” 93H 60H 33H 00H B0H sn bn le
```

Positive response

```
“P” 93H 60H 33H St1 St0 rdata
```

Negative response

```
“N” 93H 60H 33H e1 e0
```

Read block and sequence blocks from RF card

- sn(1 byte): sector number
- bn(1 byte): block number
- le(1 byte): block number (le=01H read one block, le=03H read three blocks)
- rdata(2 byte): return data

(positive response with data 9000H, and negative response with “sw1+sw2”)

Notes:
1. Ultralight Card only have one block in one sector, every block have 4 byte data. S50, S70 have 16 byte data in one block.
2. Ultralight Card, Mifare 1k (S50), Mifare 1k (S70) card range of capacity is shown as below:
   - Ultralight Card: sn=00H-0FH, bn=00H, le=01H-0FH
   - Mifare 1k (S50): sn=00H-0FH, bn=00H-03H, le=01H-04H
   - Mifare 1k (S70): sn=00H-20H, bn=00H-03H, le=01H-04H
   - sn=21H-27H, bn=00H-0FH, le=01H-10H (S70 card last 8 sector have 16 blocks)
4.6 Write sector data

Command

```
“C” 93H 60H 33H 00H D1H sn bn lc wdata
```

Positive response

```
“P” 93H 60H 33H St1 St0 rdata
```

Negative response

```
“N” 93H 60H 33H e1 e0
```

Read block and sequence blocks from RF card

- sn: sector number
- bn: block number
- le: block number
- wdata: block to write (n byte)
- rdata: return data

(positive response with data 9000H, and negative response with “sw1+sw2”)

Notes:
1. Ultralight Card only have one block in one sector, every block have 4 byte data.
   S50, S70 have 16 byte data in one block
2. Ultralight Card, Mifare 1k (S50), Mifare 1k (S70) card card range of capacity is shown as below:
   - Ultralight Card: sn=00H-0FH, bn=00H-03H, lc=01H-03H
   - Mifare 1k (S50): sn=00H-0FH, bn=00H-03H, lc=01H-03H
   - Mifare 1k (S70): sn=00H-20H, bn=00H-03H, lc=01H-03H
   - sn=21H-27H, bn=00H-0FH, lc=01H-0FH
   (S70 card last 8 sector have 16 blocks)
3. S50, S70 card last block of each sector is control sector to preserve Key A, read/write control words, Key B.

Cautions: Do not write last block and CRT-571 also will prohibit to write last block.
4.7 Initialization

Command

```
“C” 93H 60H 33H 00H D2H sn bn lc wdata
```

Positive response

```
“P” 93H 60H 33H St1 St0 rdata
```

Negative response

```
“N” 93H 60H 33H e1 e0
```

Initialization operation to RF card
- sn (1 byte): sector number
- bn (1 byte): block number
- lc (1 byte): length lc=04H
- wdata: data (4 byte)
- rdata (2 byte): return data

(positive response with data 9000H, and negative response with “sw1+sw2”)

Notes: Mifare 1k(S50), Mifare 1k (S70) card operation sector
(Sector cannot be out of range and last block can not be operated)

Mifare 1k(S50): sn=00H-0FH, bn=00H-03H,
Mifare 1k(S70): sn=00H-20H, bn=00H-03H,
    sn=20H-27H, bn=00H-0EH,
(S70 card last 8 sector have 16 blocks)
**4.8 Read value**

**Command**

```
“C” 93H 60H 33H 00H B1H sn bn
```

**Positive response**

```
“P” 93H 60H 33H St1 St0 rdata
```

**Negative response**

```
“N” 93H 60H 33H e1 e0
```

Read value operations to RF card
- sn(1 byte): sector number
- bn(1 byte): block number
- rdata(2 byte): return data

(positive response with data 9000H, and negative response with “sw1+sw2”)

**Notes:**
- Mifare 1k(S50), Mifare 1k (S70) card operation sector
- (Sector can not be out of range and last block can not be operated)

Mifare 1k(S50): sn=00H-0FH, bn=00H-03H,
Mifare 1k(S70): sn=00H-20H, bn=00H-03H,
    sn=20H-27H, bn=00H-0EH,
(S70 card last 8 sector have 16 blocks)
4.9 Increment

Command

```
“C” 93H 60H 33H 00H D3H sn bn lc wdata
```

Positive response

```
“P” 93H 60H 33H St1 St0 rdata
```

Negative response

```
“N” 93H 60H 33H e1 e0
```

Increment operation to RF card

- sn (1 byte): sector number
- bn (1 byte): block number
- lc (1 byte): increment length lc=04H
- wdata: increment data (4 byte)
- rdata (2 byte): return data

(positive response with data 9000H, and negative response with “sw1+sw2”)

Notes:
- Mifare 1k(S50), Mifare 1k (S70) card operation sector
- (Sector can not be out of range and last block can not be operated)
  - Mifare 1k(S50): sn=00H-0FH, bn=00H-03H,
  - Mifare 1k(S70): sn=00H-0FH, bn=00H-03H,
  - sn=20H-27H, bn=00H-0EH,

(S70 card last 8 sector have 16 blocks)
4.10 Decrement

Command

```
“C” 93H 60H 33H 00H D4H sn bn lc wdata
```

Positive response

```
“P” 93H 60H 33H St1 St0 rdata
```

Negative response

```
“N” 93H 60H 33H e1 e0
```

Decrement operation to RF sector

- sn(1 byte): sector number
- bn(1 byte): block number
- lc(1 byte): Decrement length lc=04H
- wdata: Decrement data(4 byte)
- rdata(2 byte): return data

(positive response with data 9000H, and negative response with “sw1+sw2”)

Notes: Mifare 1k(S50), Mifare 1k (S70) card operation sector
(Sector can not be out of range and last block can not be operated)

- Mifare 1k(S50): sn=00H-0FH, bn=00H-03H,
- Mifare 1k(S70): sn=00H-20H, bn=00H-03H,
  sn=20H-27H, bn=00H-0EH,

(S70 card last 8 sector have 16 blocks)
4.11 Type A RF card communication
Command

```
"C" 93H 60H 34H C-APDU
```

Positive response

```
"P" 93H 60H 34H St1 St0 R-APDU
```

Negative response

```
"N" 93H 60H 34H e1 e0
```

This exchanges data between RF card by protocol RF Type A T=CL according to ISO/IEC 14443-4

Notes: The max. length of C-APDU is 261 byte, the max. length of R-APDU is 258 byte.

4.12 Type B RF card communication
Command

```
"C" 93H 60H 35H C-APDU
```

Positive response

```
"P" 93H 60H 35H St1 St0 R-APDU
```

Negative response

```
"N" 93H 60H 35H e1 e0
```

This exchanges data between RF card by protocol RF Type B T=CL according to ISO/IEC 14443-4

Notes: The max. length of C-APDU is 261 byte, the max. length of R-APDU is 258 byte.
Annex IV Explanation of error code

Every error status can be cleared by procedure of (Re-Start by Initialize to complete normal). In this case, uses Status request command and confirm before next step that no error code remain.

1. Error in communication and card inject

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;00&quot;</td>
<td>To shows that received command was undefined.</td>
<td>Cleared by receiving correct commands.</td>
</tr>
<tr>
<td>&quot;01&quot;</td>
<td>To show command parameter error.</td>
<td>Cleared by receiving command with correct parameter.</td>
</tr>
<tr>
<td>&quot;02&quot;</td>
<td>The reception of the command that impossible to implementation.</td>
<td>Cf. Receiving read command while card is not staying inside the ICRW.</td>
</tr>
<tr>
<td>&quot;04&quot;</td>
<td>To show that error data was included in command.</td>
<td>Cleared by receiving executable command.</td>
</tr>
<tr>
<td>&quot;05&quot;</td>
<td>To show that a command, which requires the card movement, was executed before IC contacts were released from the card.</td>
<td>Execute IC contacts release command.</td>
</tr>
<tr>
<td>&quot;10&quot;</td>
<td>To show that the card was not carried to the specific location after specified number of trial for specified time duration during execution of command of carrying card in various ways.</td>
<td>Cleared in case card is fed to specified location by the repeated command. Or, cleared when the card is taken out from ICRW manually. Confirm the recovery by Status request command in this case.</td>
</tr>
<tr>
<td>&quot;13&quot;</td>
<td>To show that the card longer than 92mm is inserted into ICRW.</td>
<td>Cleared in case card is returned to card gate by eject command. (**Do not use CAPTURE command to eject the card)</td>
</tr>
<tr>
<td>&quot;14&quot;</td>
<td>To show that the card shorter than 78mm is inserted into ICRW.</td>
<td>Cleared in case card is returned to card gate by eject command. (**Do not use CAPTURE command to eject the card)</td>
</tr>
</tbody>
</table>
### List 2

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;16&quot;</td>
<td>To show that card staying inside the ICRW was moved up to the point where status request information changes. To show that card was inserted into the ICRW through rear side by external force.</td>
<td>Cleared in the case card is ejected to the rear side of ICRW by CAPTURE command.</td>
</tr>
<tr>
<td>&quot;17&quot;</td>
<td>To show that the card was not carried to the specific location after specified number of trial for specified time duration during execution of RETRIEVE command.</td>
<td>Cleared in case card is fed to specified location by the repeated command. Or, cleared when the card is taken out from ICRW manually. Confirm the recovery by Status request command in this case.</td>
</tr>
<tr>
<td>&quot;40&quot;</td>
<td>To show that the card was pulled out from ICRW through entrance gate when CAPTURE command is being executed.</td>
<td>Error code &quot;40&quot; is returned against only that CAPTURE command and does not exist.</td>
</tr>
<tr>
<td>&quot;45&quot;</td>
<td>ICRW lost sight of the card when ICRW completed to carry the card to the rear position during the card acceptance, and ICRW ejected the card to entrance gate.</td>
<td>Cleared when the card is taken out from ICRW manually. Do not use CAPTURE command to eject the card. Check if the card is normal. If the card is normal, there could be a problem on ICRW and checking ICRW is required.</td>
</tr>
<tr>
<td>&quot;46&quot;</td>
<td>- To show that the ejected card has not been withdrawn during execution of Monitoring for removal command. - To show that the time out occurred during execution of Intake/Withdraw command.</td>
<td>Error code &quot;46&quot; is returned against only that command and does not exist.</td>
</tr>
</tbody>
</table>
### 2. Error in card moving inside ICRW

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Clear</th>
</tr>
</thead>
</table>
| "11" | 1. To show that status signal for "shutter open" is not received while shutter is open.  
     | 2. When the card is within ICRW the shutter is opened forcibly. At this time, ICC and all the SAM are deactivated. | 1. Cleared when any of outer force preventing shutter operation is eliminated.  
     |                                                            | When the error is arising from mechanism failure of ICRW, cleared by mechanism adjustment.  
     |                                                            | 2. Remove the factor that the shutter is opened forcibly, and eject the card by Initialize command or Card carry command. |
| "12" | To show that Sensor is damaged or more that one card is inside ICRW.     | Check if the card is normal. If the card is normal, there could be a problem on ICRW and checking ICRW is required. |
| "15" | To show that data in EEPROM is damaged.                                 | Cleared by either of re-writing EEPROM data or replacing PCB.        |
| "18" | To show that shutter open/close detection sensor (SW2) and card width check sensor (SW1) are not operating correctly. | Cleared when any of outer force preventing sensor operation is eliminated.  
     |                                                            | When the error is arising from mechanism failure of ICRW, cleared by mechanism adjustment. |
| "19" | To show that a card was not inserted from the rear, even if 10 seconds had passed after the execution of BACK ENTRY command. | Cleared the moment this error code has been transmitted. Card is to be inserted from the rear within 10 seconds after the execution of BACK ENTRY command. |
| "51" | To show that Motor error has happened, through start / stop check in Initialize command. | Cleared when any of outer force preventing motor operation is eliminated.  
     |                                                            | When the error is arising from mechanism failure of ICRW, cleared by mechanism adjustment. |
### 3. Error in reading / writing MAG. card

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;21&quot;</td>
<td>To show that read error has happened in Multiple magnetic read command.</td>
<td>The detail of the error for each track will be showed in the response. Note: Read is retried up to 6 times automatically (including initial read).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To show that other read error except &quot;23&quot; and &quot;24&quot; has happened in Mag-Track Digital Decode read command.</td>
</tr>
<tr>
<td>&quot;20&quot;</td>
<td>To show that track has parity error.</td>
<td></td>
</tr>
<tr>
<td>&quot;23&quot;</td>
<td>To show that only SS, ES, LRC are contained in the track (no retry).</td>
<td></td>
</tr>
<tr>
<td>&quot;24&quot;</td>
<td>To show that the card has no magnetic track (no retry).</td>
<td></td>
</tr>
<tr>
<td>&quot;26&quot;</td>
<td>To show that the track has no SS.</td>
<td></td>
</tr>
<tr>
<td>&quot;27&quot;</td>
<td>To show that the track has no ES.</td>
<td></td>
</tr>
<tr>
<td>&quot;28&quot;</td>
<td>To show that the track has LRC error.</td>
<td></td>
</tr>
<tr>
<td>&quot;53&quot;</td>
<td>To show that read error in Digital Decode Read command. When there is abnormality in one of data, ICRW tries decoding with the reverse calculation from LRC. In the case that decoding succeeded as a result, ICRW transmits a negative response of the error cord &quot;53&quot; that added the data that was decoded.</td>
<td></td>
</tr>
<tr>
<td>&quot;22&quot;</td>
<td>To show that write error is detected through write / verify procedure.</td>
<td>Write routine. Write: Write (forward) - Verifying read (backward) * The above one routine is completed by one round trip of card. Verifying read: To be completed by comparison between write data and read data.</td>
</tr>
<tr>
<td></td>
<td>Notes: Repeat write routine up to 2 times.</td>
<td></td>
</tr>
<tr>
<td>&quot;25&quot;</td>
<td>To show that quality error (Jitter, Preamble, Post amble) has happened in write verify.</td>
<td></td>
</tr>
<tr>
<td>&quot;29&quot;</td>
<td>To show that the discordance of write data has happened in write verify.</td>
<td></td>
</tr>
</tbody>
</table>
4. **Other error codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;30&quot;</td>
<td>To show that power down (or power cut in short instant) is detected (or being detected). It is to be recognized as normal power down if back up power supply goes down below +12V.</td>
<td>Cleared when specified card handling procedure is completed after power recovery.</td>
</tr>
<tr>
<td>&quot;31&quot;</td>
<td>To show that DSR signal was turned to OFF (communication is cut).</td>
<td>Cleared by turning DSR signal ON.</td>
</tr>
<tr>
<td>&quot;B0&quot;</td>
<td>Received the other command before executing Initialize command.</td>
<td>Execute Initialize command.</td>
</tr>
<tr>
<td>&quot;41&quot;</td>
<td>Failure at IC Contact solenoid or sensor ICD</td>
<td>Cleared when any of outer force preventing IC Contact operation is eliminated. When the error is arising from mechanism failure of ICRW, cleared by mechanism adjustment.</td>
</tr>
<tr>
<td>&quot;43&quot;</td>
<td>Card could not be set to IC contact position / Failure at sensor PDI</td>
<td>When the error is arising from electrical failure of ICRW, cleared by electrical failure.</td>
</tr>
<tr>
<td>&quot;50&quot;</td>
<td>Retract counter overflow.</td>
<td>Set counter value ( 0 - 99 ) in Retract counter.</td>
</tr>
<tr>
<td>&quot;70&quot;</td>
<td>Failure at F-ROM operation.</td>
<td>Replace PCB.</td>
</tr>
<tr>
<td>&quot;71&quot;</td>
<td>firmware of User program code area is wrong.</td>
<td>Execute the download.</td>
</tr>
</tbody>
</table>
## 5. Error on IC card handling

### List 1

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;60&quot;</td>
<td>To show that there found abnormal condition on the power-line (Vcc) of ICC/SAM. ICRW disconnects ICC/SAM automatically.</td>
<td>- Metal card (Ask customer to remove the card.)&lt;br&gt;- Defect chip card (Ask customer to remove the card.)&lt;br&gt;- Vcc short to GND (Ask customer to remove the card and repair.)</td>
</tr>
<tr>
<td>&quot;61&quot;</td>
<td>The receiving error of ATR. ICRW has disconnected already.&lt;br&gt;- No ATR (TS is not received from 380 clock cycle and 42000 clock cycle after time set RST to High).&lt;br&gt;- Parity error on ATR.&lt;br&gt;- ATR interval time between two consecutive characters is over 9600etus.&lt;br&gt;- ATR duration over 19200etus.&lt;br&gt;- TCK error on ATR.&lt;br&gt;- ATR length is longer than 64 byte (not include TS).&lt;br&gt;- TS is neither 3FH nor 3BH</td>
<td>- No chip card. (Ask customer to remove the card)&lt;br&gt;- Defect chip card. (Ask customer to remove the card.)&lt;br&gt;- Contact broken. (Ask customer to remove the card and repair.)</td>
</tr>
<tr>
<td>&quot;62&quot;</td>
<td>To show that the specified protocol does not agree with that of ICC/SAM. ICRW still connected.&lt;br&gt;- HOST use &quot;CI4&quot; or &quot;CI5&quot; command with T=0 protocol IC card.&lt;br&gt;- HOST use &quot;CI3&quot; command with T=1 protocol IC card.&lt;br&gt;- HOST use &quot;CID&quot; or &quot;CIE&quot; command with T=0 protocol SAM.&lt;br&gt;- HOST use &quot;CIC&quot; command with T=1 protocol SAM.</td>
<td>HOST should use correct command, &quot;CI9&quot; or &quot;CII&quot;</td>
</tr>
<tr>
<td>&quot;63&quot;</td>
<td>In case T=1 cards, after ATR receiving, IFS exchange is failed. ICRW detects time out. During communication with IC card, ICRW detects time out (WT, CWT or BWT). ICRW disconnected the IC card.</td>
<td>Defect chip card. (Ask customer to remove the card.)&lt;br&gt;Non ISO-standard card. IFSresp receiving error (Ask customer to remove the card.)</td>
</tr>
<tr>
<td>Code</td>
<td>Meaning</td>
<td>Clear:</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| "64" | In case T=1 cards, after ATR receiving, IFS exchange is failed. ICRW detects protocol error. To show that there is protocol error. (other than "63")  
T=0 IC card:  
- 5 parity errors in received from IC card.  
- Status byte error (SW1 is different from 6xH or 9xH).  
- 5 parity errors in transmitting mode to IC card.  
- Procedure byte error (Procedure bytes is different from INS, Not INS, 60H, 61H or 6CH).  
T=1 IC card:  
- Bad NAD (NAD is different from 00H)  
- Bad PCB  
- Bad EDC  
- Parity error                                                                 | Defect chip card. (Ask customer to remove the card.) |
| "65" | HOST tried to communicate with IC card without card activation. ICRW has disconnected already.                                                                 | HOST should activate ICC/SAM before communication.                        |
| "66" | ICRW tried to activate with ICC/SAM, but the card returned ATR, which is not supported.  
ICRW has disconnected already.  
This error is returned with "CI03", "CI05" or "CI06".  
This error is returned with "CI@3", "CI@5" or "CI@6".                                                                 | Ask customer to remove the card.                              |
| "69" | ICRW tried to activate with ICC/SAM, but the card returned ATR, which does not match EMV.  
ICRW has disconnected already.  
This error is returned with "CI0", "CI00".  
This error is returned with "CI@", "CI@0".                                                                 | HOST tries other activate command "CI03", "CI05" or "CI06". Or Ask customer to remove the card. |
## Annex V  Values of ATR parameter (TA1 and TA2)

### Table1: Supportable TA1 values

<table>
<thead>
<tr>
<th>Vcc</th>
<th>Condition</th>
<th>Support</th>
<th>Communication speed (F,D)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(Yes/No)</td>
<td></td>
</tr>
<tr>
<td>30H</td>
<td>TA1 = 'any' and TA2=none</td>
<td>Yes</td>
<td>9622bps (F=372, D=1)</td>
</tr>
<tr>
<td></td>
<td>TA1 = 'any' and TA2.b5 = 0</td>
<td>Yes (*1)</td>
<td>Comply with Table3</td>
</tr>
<tr>
<td></td>
<td>TA2.b5=1</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>33H</td>
<td>TA1 = '11' and TA2=none</td>
<td>Yes</td>
<td>9622bps (F=372, D=1)</td>
</tr>
<tr>
<td>35H</td>
<td>TA1 = 'any' and TA2=none</td>
<td>Yes</td>
<td>If TA1 is shown in Table2, ICRW sends PPS request.</td>
</tr>
<tr>
<td></td>
<td>(Not including TA1=’11’)</td>
<td></td>
<td>Communication speed depends on PPS response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If TA1 is not shown in Table2, ICRW does not sends PPS request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Communication speed is 9622bps (F=372, D=1).</td>
</tr>
<tr>
<td></td>
<td>TA1='any' and TA2.b5=0</td>
<td>Yes(*1)</td>
<td>Comply with Table3</td>
</tr>
<tr>
<td></td>
<td>TA1='any' and TA2.b5=1</td>
<td>Yes</td>
<td>9622bps (F=372, D=1)</td>
</tr>
<tr>
<td>40H</td>
<td>TA1 = 'any' and TA2=none (Including TA1='11')</td>
<td>Yes</td>
<td>9622bps (F=372, D=1)</td>
</tr>
</tbody>
</table>

(*1) When TA1 exists in Table3, ICRW supports its TA1.
Table2: TA1 values that ICRW sends PPS request.

| TA1 | 02, 12, 03, 13, 32, 33, 53, 54, 92, 93, B2, B3, D3, D4 |

A meaning of Vcc parameter please refer "activate ICC command".

Table3: Supported TA1 values in specific mode.

<table>
<thead>
<tr>
<th>D=</th>
<th>F=</th>
<th>1 (9622)</th>
<th>2 (19244)</th>
<th>4 (38490)</th>
<th>8</th>
<th>16</th>
<th>CLK frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>372</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.58MHz</td>
</tr>
<tr>
<td>372</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.58MHz</td>
</tr>
<tr>
<td>558</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>744</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.16MHz</td>
</tr>
<tr>
<td>1116</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1488</td>
<td>-</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>-</td>
<td>-</td>
<td>7.16MHz</td>
</tr>
<tr>
<td>1860</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.58MHz</td>
</tr>
<tr>
<td>768</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.16MHz</td>
</tr>
<tr>
<td>1536</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>2048</td>
<td>-</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>-</td>
<td>-</td>
<td>7.16MHz</td>
</tr>
</tbody>
</table>

Upper row: TA1 value
(Lower row): Communication speed (bps)
Table 4: Supported values of ATR

<table>
<thead>
<tr>
<th>ATR</th>
<th>Vcc</th>
<th>30H</th>
<th>33H</th>
<th>35H</th>
<th>36H</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Supported values</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS</td>
<td>“3F”, “3B”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA1</td>
<td>See Table 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB1</td>
<td>“00” (cold reset) any value (warm reset) (*1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC1</td>
<td>any value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TD1</td>
<td>m.s. nibble: any value l.s. nibble: ‘0’ or ‘1’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA2</td>
<td>See Table 1 and TA2 l.s. nibble = TD1 l.s. nibble</td>
<td>See Table 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB2</td>
<td>None (prohibit)</td>
<td>any value (*1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>“01”…”FF”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TD2</td>
<td>any value l.s. nibble: ‘1’, ‘E’ m.s. nibble: any value l.s. nibble: any value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOT</td>
<td>T=15 ‘10’…”FE’ ‘01’…”FE’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA3,TA4</td>
<td>m.s. nibble: ‘0’…’4’ and l.s. nibble: 0’…’5’ and (2CW1 &gt; (N+1))</td>
<td>m.s. nibble: “0”…”9” and l.s. nibble: “0”…”15” and (2CW1 &gt; (N+1))</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB3,TB4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TC3,TC4</td>
<td>“00” any value</td>
<td>any value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TD3,TD4</td>
<td>any value any value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T=15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB3,TC3, TD3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA4</td>
<td>b1=1</td>
<td>b1=1</td>
<td>b2=1 or b1=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB4,TC4</td>
<td>any value</td>
<td>any value</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A meaning of Vcc parameter please refer "activate ICC command".

(*1) ICRW does not generate Vpp.

(*2) ‘F’(T=15) is prohibited in TD2 l.s.nibble.
Annex VI C-APDU Format

The C-APDU consists of a mandatory header of four consecutive bytes denoted CLA, INS, P1 and P2, followed by a conditional body of variable length. The meanings of every byte are below.

<table>
<thead>
<tr>
<th>byte</th>
<th>meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLA</td>
<td>Instruction Class</td>
</tr>
<tr>
<td>INS</td>
<td>Instruction Code</td>
</tr>
<tr>
<td>P1</td>
<td>Instruction Parameter 1</td>
</tr>
<tr>
<td>P2</td>
<td>Instruction Parameter 2</td>
</tr>
<tr>
<td>Lc</td>
<td>Byte Length of Data Field</td>
</tr>
<tr>
<td>Data</td>
<td>Data Field</td>
</tr>
<tr>
<td>Le</td>
<td>Byte Length of Expected Response Length</td>
</tr>
</tbody>
</table>

About the details of each byte, refer to specifications of every card's standard.

The C-APDU structure has following four cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CLA INS P1 P2</td>
</tr>
<tr>
<td>2</td>
<td>CLA INS P1 P2 Le</td>
</tr>
<tr>
<td>3</td>
<td>CLA INS P1 P2 Lc Data</td>
</tr>
<tr>
<td>4</td>
<td>CLA INS P1 P2 Lc Data Le</td>
</tr>
</tbody>
</table>

The host shall transmit the command of Case1, Case2, Case3 and Case4 correctly. Especially for the case 1 on T=0 protocol, ICRW adds '00' internally as the fifth byte of the command to the card.
Annex VII Calculation method of CRCC

CRCC(X16+X12+X5+1) is made by the following method.

/* [data]
   hex 0xF2, 0x00, 0x08, 0x43, 0x30, 0x30, 0x33, 0x32, 0x34, 0x30, 0x30
   CRC 0xFACE
*/
#define INIT 0x0000 /* Initial value */
#define POLINOMIAL 0x1021 /* Polynomial X16+X12+X5+1 */

unsigned short calc_crc(unsigned short crc, unsigned short ch);
unsigned short GetCRC(unsigned char *p, unsigned short n);

unsigned short calc_crc(unsigned short crc, unsigned short ch)
{
    unsigned short i;
    ch <<= 8;
    for (i = 8; i > 0; i--) {
        if ((ch ^ crc) & 0x8000) {
            crc = (crc << 1) ^ POLINOMIAL;
        } else {
            crc <<= 1;
        }
        ch <<= 1;
    }
    return crc;
}

/* Generate GetCRC */
unsigned short GetCRC(unsigned char *p, unsigned short n)
{
    unsigned char ch;
    unsigned short i;
    unsigned short crc = INIT;

    for (i = 0; i < n; i++) {

        ch = *p++;
        crc = calc_crc(crc, (unsigned short)ch);
    }
}
int main(void)
{
    /* Transmission command
    STX : F2H
    LEN : 00 08H
    TEXT: Initialize command ("C0032400")
    */
    unsigned char TransCommand[13] = {0xF2,0x00,0x08,0x43,0x30,0x30,0x33,0x32,0x34,0x30,0x30,0x00,0x00};
    unsigned short TextLength = 11; /* length of (STX+LEN+TEXT) */
    unsigned short crc; /* CRC */
    crc = GetCRC(TransCommand, TextLength);
    TransCommand[11] = (crc >> 8) & 0xFF;
    TransCommand[12] = crc & 0xFF;

    return 0;
}

7.1 Format

<table>
<thead>
<tr>
<th>Track Information</th>
<th>ISO#1</th>
<th>ISO#2</th>
<th>ISO#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character length</td>
<td>7bits</td>
<td>5bits</td>
<td>5bits</td>
</tr>
<tr>
<td>Parity</td>
<td>Odd</td>
<td>Odd</td>
<td>Odd</td>
</tr>
<tr>
<td>Longitudinal redundancy check (LRC)</td>
<td>Even SS - ES</td>
<td>Even SS - ES</td>
<td>Even SS - ES</td>
</tr>
<tr>
<td>Start sentinel (SS)</td>
<td>&quot;%&quot;</td>
<td>&quot;;&quot;</td>
<td>&quot;;;&quot;</td>
</tr>
<tr>
<td>End sentinel (ES)</td>
<td>&quot;?&quot;</td>
<td>&quot;?&quot;</td>
<td>&quot;?&quot;</td>
</tr>
<tr>
<td>Data number (min-max)</td>
<td>1 - 76</td>
<td>1 - 37</td>
<td>1 - 104</td>
</tr>
</tbody>
</table>

7.2 Coded character set for ISO#1

The 14 characters ! " & * + , : ; < = > @ _ are available for hardware control purposes and may not be used for information (data content).

The 3 characters [ £ □ ] are reserved for additional national characters when required. They must not be used internationally.

The character # is reserved for optional additional graphic symbols.

The 3 characters % ^ ? shall have the following meaning:

- % start sentinel
- ^ field separator
- ? end sentinel
### Communication Protocol

#### Card Reader/Writer

<table>
<thead>
<tr>
<th>Char.</th>
<th>Binary</th>
<th>Char.</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>2^6</td>
<td>2^5</td>
<td>2^4</td>
</tr>
<tr>
<td>space</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>!</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>#</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>%</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&amp;</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>'</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>*</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>,</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>:</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>;</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&lt;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>=</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&gt;</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Note: ICRW can read or write the data except SS and ES.
7.3 Coded character set for ISO#2,3

The 3 characters : < > are available for hardware control purposes and may not be used for information (data content).

The 3 characters; = ? shall have the following meaning:
- ; start sentinel
- = field separator
- ? end sentinel

<table>
<thead>
<tr>
<th>Char.</th>
<th>Binary P</th>
<th>2&lt;sup&gt;3&lt;/sup&gt;</th>
<th>2&lt;sup&gt;2&lt;/sup&gt;</th>
<th>2&lt;sup&gt;1&lt;/sup&gt;</th>
<th>2&lt;sup&gt;0&lt;/sup&gt;</th>
<th>Char.</th>
<th>Binary P</th>
<th>2&lt;sup&gt;3&lt;/sup&gt;</th>
<th>2&lt;sup&gt;2&lt;/sup&gt;</th>
<th>2&lt;sup&gt;1&lt;/sup&gt;</th>
<th>2&lt;sup&gt;0&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>;</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>;</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>&lt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>=</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>&gt;</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The end of the document.
<table>
<thead>
<tr>
<th>Communication Protocol</th>
<th>Model No.</th>
<th>CRT-350</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>2010/1/19</td>
<td></td>
</tr>
<tr>
<td>Ver.</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>155/155</td>
<td></td>
</tr>
</tbody>
</table>