# EMV®
# Level 1 Specifications for Payment Systems

## EMV Contact Interface Specification

Version 1.0
October 2022

## Legal Notice

The EMV® Specifications are provided "AS IS" without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of the EMV® Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of the EMV® Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party's infringement of any intellectual property rights in connection with the EMV® Specifications.

# Revision Log - Version 1.0

The following changes have been made to Book 1 since the publication of Version 4.3. Book 1 has been split and this new document addresses only the Electromechanical Characteristics, Logical Interface and Transmission Protocols and is now part of the EMV Level 1 Specifications for Payment Systems. According to the General Bulletin GB No 49, the version numbering scheme has been set to Version 1.0.

The Legal Notice and the Copyright has been updated.

**Incorporated changes described in the following Specification Bulletins:**

Specification Bulletin No. 114: Case 4 Command Processing with Warning Condition

General Bulletin No. 11, Third Edition: Lower Voltage Card Migration Mandatory Implementation Schedule

Specification Bulletin No. 202: Contact Card – Transients on Current drawn from VCC

Specification Bulletin No. 216, Third Edition: Contact – Technology removal for Vpp

Specification Bulletin No. 218, Second Edition: Contact – ATR parameter requirements

Specification Bulletin No. 219, Second Edition: Contact – Terminal clock signal frequency (CLK)

Specification Bulletin No. 247, Second Edition: Contact – Communication Performance Enhancement

Specification Bulletin No. 264: Contact – Warm Reset Initiation

Specification Bulletin No. 265, Second Edition: Contact – TCK Check Character Requirement in ATR

**As well as**

Sections 5.1 and 5.3.6 were redrafted and aligned with General Bulletin 11 and Section 5.1 of version 4.3.

Sections 5.3.2.1, 5.3.2.2, 5.3.4, 5.3.5: Table for class A only ICC was removed.

Section 5.3.4: Note removed since already in section 5.5.4 at the bottom of Table 13.

Section 9.4: Note removed as not relevant to this specification.

Furthermore, numbering and cross references in this version have been updated to reflect changes introduced by the published bulletins as well as minor editorials such as corrections of typos and syntax. The Abbreviations section has been updated by keeping only the relevant definitions. The Normative References section has been updated by keeping only the relevant ISO Standards. The Definitions section has been updated by keeping only the relevant definitions. The Index section has been removed.

# Contents

# Figures

# Tables

# 1 Scope and Audience

This specification is intended for use by manufacturers of ICCs and terminals, system designers in payment systems, and financial institution staff responsible for implementing financial applications in ICCs.

This document, the *EMV Level 1 Specifications for Payment Systems - EMV Contact Interface Specification* describes the minimum functionality required of integrated circuit cards (ICCs) and terminals to ensure correct operation and interoperability independent of the application to be used. Additional proprietary functionality and features may be provided, but these are beyond the scope of this specification and interoperability cannot be guaranteed.

The *EMV Level 1 Specifications for Payment Systems* includes the following documents, all available on http://www.emvco.com:

- *EMV Contact Interface Specification*
- *EMV Contactless Interface Specification*

## 1.1 Changes in Version 1.0

This release incorporates all relevant Specification Bulletins, Application Notes, amendments, etc. published up to the date of this release.

The Revision Log at the beginning of the Book provides additional detail about changes to this specification.

## 1.2 Underlying Standards

This specification is based on the ISO/IEC 7816 series of standards and should be read in conjunction with those standards. However, if any of the provisions or definitions in this specification differ from those standards, the provisions herein shall take precedence.

# 2 Normative References

The following standards contain provisions that are referenced in these specifications. The latest version shall apply unless a publication date is explicitly stated.

| | |
|---|---|
| ISO/IEC 7811-1 | Identification cards – Recording technique – Part 1: Embossing |
| ISO/IEC 7813 | Identification cards – Financial transaction cards |
| ISO/IEC 7816-1 | Identification cards – Integrated circuit(s) cards with contacts – Part 1: Physical characteristics |
| ISO/IEC 7816-2 | Information technology – Identification cards – Integrated circuit(s) cards with contacts – Part 2: Dimensions and location of contacts |
| ISO/IEC 7816-3 | Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols |
| ISO/IEC 7816-4 | Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange |
| ISO/IEC 10373 | Identification cards – Test methods |

# 3 Definitions

The following terms are used in this specification.

| | |
|---|---|
| **Application** | The application protocol between the card and the terminal and its related set of data. |
| **Block** | A succession of characters comprising two or three fields defined as prologue field, information field, and epilogue field. |
| **Byte** | 8 bits. |
| **Card** | A payment card as defined by a payment system. |
| **Cold Reset** | The reset of the ICC that occurs when the supply voltage (VCC) and other signals to the ICC are raised from the inactive state and the reset (RST) signal is applied. |
| **Command** | A message sent by the terminal to the ICC that initiates an action and solicits a response from the ICC. |
| **Contact** | A conducting element ensuring galvanic continuity between integrated circuit(s) and external interfacing equipment. |
| **Deactivation Sequence** | The deactivation sequence defined in section 6.1.5. |
| **Embossing** | Characters raised in relief from the front surface of a card. |
| **Epilogue Field** | The final field of a block. It contains the error detection code (EDC) byte(s). |
| **Exclusive-OR** | Binary addition with no carry, giving the following values:<br>0 + 0 = 0<br>0 + 1 = 1<br>1 + 0 = 1<br>1 + 1 = 0 |
| **Guardtime** | The minimum time between the trailing edge of the parity bit of a character and the leading edge of the start bit of the following character sent in the same direction. |

| | |
|---|---|
| **Inactive** | The supply voltage (VCC) and other signals to the ICC are in the inactive state when they are at a potential of 0.4 V or less with respect to ground (GND). |
| **Integrated Circuit Module** | The sub-assembly embedded into the ICC comprising the IC, the IC carrier, bonding wires, and contacts. |
| **Integrated Circuit(s)** | Electronic component(s) designed to perform processing and/or memory functions. |
| **Integrated Circuit(s) Card** | A card into which one or more integrated circuits are inserted to perform processing and memory functions. |
| **Interface Device** | That part of a terminal into which the ICC is inserted, including such mechanical and electrical devices as may be considered part of it. |
| **Magnetic Stripe** | The stripe containing magnetically encoded information. |
| **Message** | A string of bytes sent by the terminal to the card or vice versa, excluding transmission-control characters. |
| **Nibble** | The four most significant or least significant bits of a byte. |
| **Prologue Field** | The first field of a block. It contains subfields for node address (NAD), protocol control byte (PCB), and length (LEN). |
| **Response** | A message returned by the ICC to the terminal after the processing of a command message received by the ICC. |
| **Signal Amplitude** | The difference between the high and low voltages of a signal. |
| **Signal Perturbations** | Abnormalities occurring on a signal during normal operation such as undershoot/overshoot, electrical noise, ripple, spikes, crosstalk, etc. Random perturbations introduced from external sources are beyond the scope of this specification. |
| **State H** | Voltage high on a signal line. May indicate a logic one or logic zero depending on the logic convention used with the ICC. |

| | |
|---|---|
| **State L** | Voltage low on a signal line. May indicate a logic one or logic zero depending on the logic convention used with the ICC. |
| **T=0** | Character-oriented asynchronous half duplex transmission protocol. |
| **T=1** | Block-oriented asynchronous half duplex transmission protocol. |
| **Terminal** | The device used in conjunction with the ICC at the point of transaction to perform a financial transaction. The terminal incorporates the interface device and may also include other components and interfaces such as host communications. |
| **Terminate Card Session** | End the card session by deactivating the IFD contacts according to section 6.1.5, and displaying a message indicating that the ICC cannot be used to complete the transaction |
| **Transaction** | An action taken by a terminal at the user's request. For a POS terminal, a transaction might be payment for goods, etc. A transaction selects among one or more applications as part of its processing flow. |
| **Warm Reset** | The reset that occurs when the reset (RST) signal is applied to the ICC while the clock (CLK) and supply voltage (VCC) lines are maintained in their active state. |

# 4 Abbreviations, Notations, and Terminology

## 4.1 Abbreviations

| | |
|---|---|
| µA | Microampere |
| µm | Micrometre |
| µs | Microsecond |
| ACK | Acknowledgment |
| APDU | Application Protocol Data Unit |
| ATR | Answer to Reset |
| BGT | Block Guardtime |
| BWI | Block Waiting Time Integer |
| BWT | Block Waiting Time |
| C | Celsius or Centigrade |
| C-APDU | Command APDU |
| $C_{IN}$ | Input Capacitance |
| CLA | Class Byte of the Command Message |
| CLK | Clock |
| C-TPDU | Command TPDU |
| CWI | Character Waiting Time Integer |
| CWT | Character Waiting Time |
| D | Bit Rate Adjustment Factor |
| DAD | Destination Node Address |
| DC | Direct Current |
| EDC | Error Detection Code |
| etu | Elementary Time Unit |
| f | Frequency |

| | |
|---|---|
| GND | Ground |
| I/O | Input/Output |
| I-block | Information Block |
| IC | Integrated Circuit |
| ICC | Integrated Circuit(s) Card |
| $I_{CC}$ | Current drawn from VCC |
| IEC | International Electrotechnical Commission |
| IFD | Interface Device |
| IFS | Information Field Size |
| IFSC | Information Field Size for the ICC |
| IFSD | Information Field Size for the Terminal |
| IFSI | Information Field Size Integer |
| INF | Information Field |
| INS | Instruction Byte of Command Message |
| $I_{OH}$ | High Level Output Current |
| $I_{OL}$ | Low Level Output Current |
| ISO | International Organization for Standardization |
| l.s. | Least Significant |
| Lc | Exact Length of Data Sent by the TAL in a Case 3 or 4 Command |
| Le | Maximum Length of Data Expected by the TAL in Response to a Case 2 or 4 Command |
| LEN | Length |
| Licc | Exact Length of Data Available or Remaining in the ICC (as Determined by the ICC) to be Returned in Response to the Case 2 or 4 Command Received by the ICC |
| LRC | Longitudinal Redundancy Check |
| m$\Omega$ | Milliohm |
| M$\Omega$ | Megohm |
| m.s. | Most Significant |

| | |
|---|---|
| m/s | Meters per Second |
| mA | Milliampere |
| max. | Maximum |
| MHz | Megahertz |
| min. | Minimum |
| mm | Millimetre |
| N | Newton |
| NAD | Node Address |
| NAK | Negative Acknowledgment |
| nAs | Nanoampere-second |
| ns | Nanosecond |
| P1 | Parameter 1 |
| P2 | Parameter 2 |
| P3 | Parameter 3 |
| PCB | Protocol Control Byte |
| pF | Picofarad |
| POS | Point of Service |
| PPS | Protocol Parameter Selection |
| R-APDU | Response APDU |
| R-block | Receive Ready Block |
| RFU | Reserved for Future Use |
| RST | Reset |
| SAD | Source Node Address |
| S-block | Supervisory Block |
| SW1 | Status Byte One |
| SW2 | Status Byte Two |
| TCK | Check Character |
| $t_F$ | Fall Time Between 90% and 10% of Signal Amplitude |

| | |
|---|---|
| TPDU | Transport Protocol Data Unit |
| $t_R$ | Rise Time Between 10% and 90% of Signal Amplitude |
| TS | Initial Character |
| TTL | Terminal Transport Layer |
| V | Volt |
| $V_{CC}$ | Voltage Measured on VCC Contact |
| VCC | Supply Voltage |
| $V_{IH}$ | High Level Input Voltage |
| $V_{IL}$ | Low Level Input Voltage |
| $V_{OH}$ | High Level Output Voltage |
| $V_{OL}$ | Low Level Output Voltage |
| WI | Waiting Time Integer |
| WTX | Waiting Time Extension |
| WWT | Work Waiting Time |

## 4.2 Notations

| | |
|---|---|
| '0' to '9' and 'A' to 'F' | 16 hexadecimal characters |
| xx | Any value |

## 4.3 Terminology

| | |
|---|---|
| proprietary | Not defined in this specification and/or outside the scope of this specification |
| shall | Denotes a mandatory requirement |
| should | Denotes a recommendation |

# 5 Electromechanical Interface

This section covers the electrical and mechanical characteristics of the ICC and the terminal. ICC and terminal specifications differ to allow a safety margin to prevent damage to the ICC.

The ICC characteristics defined herein are based on the ISO/IEC 7816 series of standards with some small variations.

## 5.1 Lower Voltage ICC Migration

A phased migration to lower voltage cards has occurred. Cards that support class A[1] only are phased out and have been replaced by class AB or class ABC cards.

Cards shall be either class AB or class ABC.

Terminals can be either class A terminals (or the class A component of multi-class terminals) or class B (or the class B component of multi-class terminals). For proprietary reasons terminals may additionally support Class C, however, class C only terminals are not permitted.

## 5.2 Mechanical Characteristics of the ICC

This section describes the physical characteristics, contact assignment, and mechanical strength of the ICC.

### 5.2.1 Physical Characteristics

Except as otherwise specified herein, the ICC shall comply with the physical characteristics for ICCs as defined in ISO/IEC 7816-1. The ICC shall also comply with the additional characteristics defined in ISO/IEC 7816-1 as related to ultraviolet light, X-rays, surface profile of the contacts, mechanical strength, electromagnetic characteristics, and static electricity and shall continue to function correctly electrically under the conditions defined therein.

The highest point on the IC module surface shall not be greater than 0.10mm above the plane of the card surface.

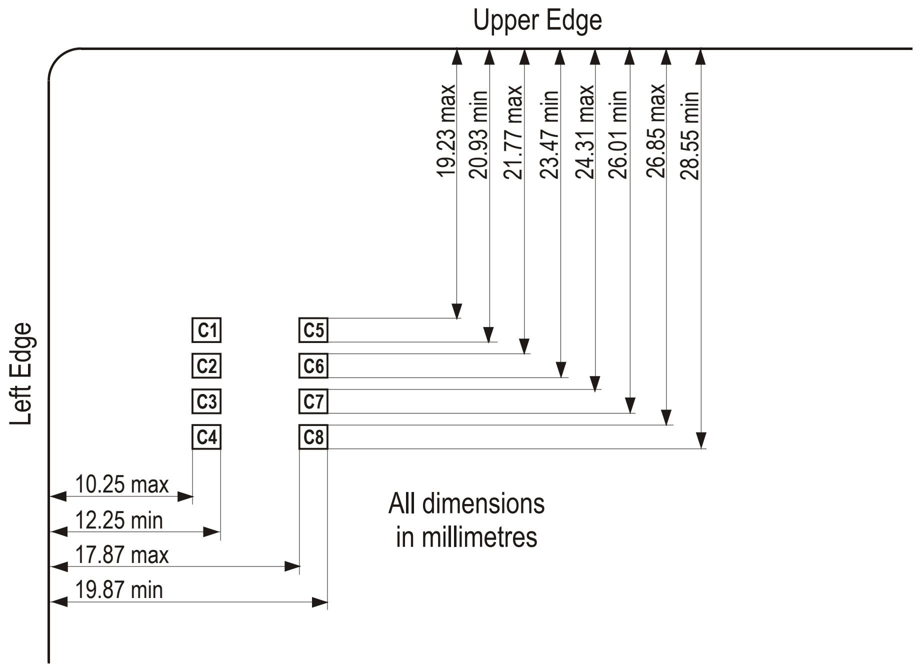
The lowest point on the IC module surface shall not be greater than 0.10mm below the plane of the card surface.

[1] A = 5 volts, B = 3 volts, C = 1.8 volts (see ISO/IEC 7816-3).

### 5.2.2 Dimensions and Location of Contacts

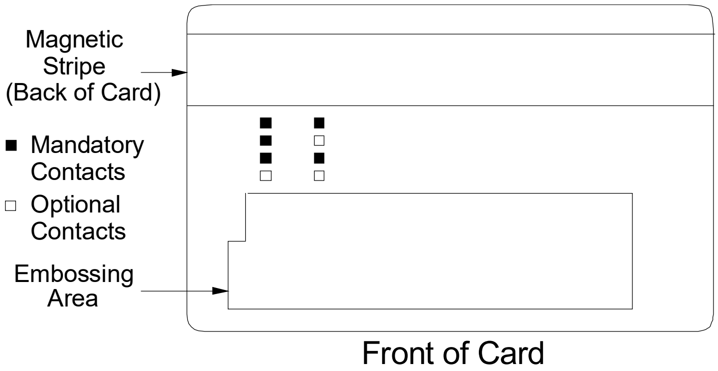
The dimensions and location of the contacts shall be as shown in Figure 1:

**Figure 1: ICC Contact Location and Dimensions**




Areas C1, C2, C3, C5, and C7 shall be fully covered by conductive surfaces forming the minimum ICC contacts. Areas C4, C6, C8, and areas Z1 to Z8 as defined in ISO/IEC 7816-2 Annex B may optionally have conductive surfaces, but it is strongly recommended that no conductive surfaces exist in areas Z1 to Z8. If conductive surfaces exist in areas C6, and Z1 to Z8, they shall be electrically isolated from the integrated circuit (IC), from one another, and from any other contact area. (Electrically isolated means that the resistance measured between the conductive surface and any other conductive surface shall be ≥10MΩ with an applied voltage of 5V DC.) In addition, there shall be no connection between the conductive surface of any area and the conductive surface of any other area, other than via the IC. The minimum ICC contacts shall be connected to the IC contacts as shown in Table 1. The layout of the contacts relative to embossing and/or magnetic stripe shall be as shown in Figure 2:

**Figure 2: Layout of Contacts**




**Note:** Care should be taken that card embossing does not damage the IC. Further, positioning of the signature panel behind the IC may lead to damage due to heavy pressure being applied during signature.

### 5.2.3 Contact Assignment

The assignment of the ICC contacts shall be as defined in ISO/IEC 7816-2 and is shown in Table 1:

**Table 1: ICC Contact Assignment**

| | | | |
|---|---|---|---|
| C1 | Supply voltage (VCC) | C5 | Ground (GND) |
| C2 | Reset (RST) | C6 | RFU |
| C3 | Clock (CLK) | C7 | Input/output (I/O) |
| C4 | Not used; need not be physically present | C8 | Not used; need not be physically present |

# 5.3 Electrical Characteristics of the ICC

This section describes the electrical characteristics of the signals as measured at the ICC contacts.

## 5.3.1 Measurement Conventions

All measurements are made at the point of contact between the ICC and the interface device (IFD) contacts and are defined with respect to the GND contact over an ambient temperature range 0° C to 50° C. ICCs shall be capable of correct operation over an ambient temperature range of at minimum 0° C to 50° C.

All currents flowing into the ICC are considered positive.

**Note:** The temperature range limits are dictated primarily by the thermal characteristics of polyvinyl chloride (which is used for the majority of cards that are embossed) rather than by constraints imposed by the characteristics of the IC.

## 5.3.2 Input/Output (I/O)

This contact is used as an input (reception mode) to receive data from the terminal or as an output (transmission mode) to transmit data to the terminal. During operation, the ICC and the terminal shall not both be in transmission mode. In the event that this condition occurs, the state (voltage level) of the I/O contact is indeterminate and no damage shall occur to the ICC.

### 5.3.2.1 Reception Mode

When in reception mode, and with the supply voltage (VCC) for the applicable class in the range specified in section 5.3.6, the ICC shall correctly interpret signals from the terminal having the characteristics shown in Table 2:

**Table 2: Electrical Characteristics of I/O for ICC Reception**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{IH}$ | | 0.7 x $V_{CC}$ | $V_{CC}$ | V |
| $V_{IL}$ | | 0 | 0.2 x $V_{CC}$ | V |
| $t_R$ and $t_F$ | | — | 1.0 | μs |
| The ICC shall not be damaged by signal perturbations on the I/O line in the range –0.3 V to $V_{CC}$ + 0.3 V. | | | | |

#### 5.3.2.2 Transmission Mode

When in transmission mode, the ICC shall send data to the terminal with the characteristics shown in Table 3:

**Table 3: Electrical Characteristics of I/O for ICC Transmission**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{OH}$ | $-20\ \mu A < I_{OH} < 0$ | $0.7 \times V_{CC}$ | $V_{CC}$ | V |
| $V_{OL}$ | Class A: $0 < I_{OL} < 1$ mA<br>Classes B and C: $0 < I_{OL} < 0.5$ mA | 0<br>0 | $0.08 \times V_{CC}$<br>$0.15 \times V_{CC}$ | V |
| $t_R$ and $t_F$ | $C_{IN\ (terminal)}$ = 30 pF max. | — | 1.0 | µs |

Unless transmitting, the ICC shall set its I/O line driver to reception mode. There is no requirement for the ICC to have any current source capability to I/O.

### 5.3.3 Contact C6

No ICC shall be damaged by an IFD where contact C6 is connected in the IFD to VCC or GND.

### 5.3.4 Clock (CLK)

With VCC in the range specified for the applicable class in section 5.3.6, the ICC shall operate correctly with a CLK signal having the characteristics shown in Table 4:

**Table 4: Electrical Characteristics of CLK to ICC**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{IH}$ | | $0.7 \times V_{CC}$ | $V_{CC}$ | V |
| $V_{IL}$ | | 0 | $0.2 \times V_{CC}$ | V |
| $t_R$ and $t_F$ | | — | 9% of clock period | |
| The ICC shall not be damaged by signal perturbations on the CLK line in the range –0.3 V to $V_{CC}$ + 0.3 V. | | | | |

The ICC shall operate correctly with a CLK duty cycle of between 44% and 56% of the period during stable operation.

The ICC shall operate correctly with a CLK frequency in the range 1 MHz to 5 MHz.

### 5.3.5 Reset (RST)

With VCC in the range specified for the applicable class in section 5.3.6, the ICC shall correctly interpret a RST signal having the characteristics shown in Table 5:

**Table 5: Electrical Characteristics of RST to ICC**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{IH}$ | | 0.7 x $V_{CC}$ | $V_{CC}$ | V |
| $V_{IL}$ | | 0 | 0.2 x $V_{CC}$ | V |
| $t_R$ and $t_F$ | $V_{CC}$ = min. to max. | — | 1.0 | µs |
| The ICC shall not be damaged by signal perturbations on the RST line in the range –0.3 V to $V_{CC}$ + 0.3 V. | | | | |

The ICC shall answer to reset asynchronously using active low reset.

### 5.3.6 Supply Voltage (VCC)

Three classes of operating conditions are defined for the ICC. These are specified in Table 6.

The ICC shall support classes A and B and may optionally support classes A, B and C.

**Table 6: Classes of Operation**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Class A<br>Class B<br>Class C | 4.50<br>2.70<br>1.62 | 5.50<br>3.30<br>1.98 | V |
| $I_{CC}$ | Class A<br>Class B<br>Class C | | 50<br>50<br>30 | mA |
| The maximum current consumptions shown apply when operating at any frequency within the range specified in section 5.3.4.<br>The current value is averaged over 1 ms. | | | | |

The ICC shall operate correctly on any supply voltage lying within the range(s) specified for the classes it supports and when operating at any frequency within the range specified in section 5.3.4.

The ICC shall not be damaged if it is operated under classes that it does not support (the ICC is considered to be damaged if it no longer operates as specified, or if it contains corrupt data).

**Table 7: Mandatory and Optional Operating Voltage Ranges**

| Supported Classes | ICC Shall Operate | ICC May Operate | Unit |
|---|---|---|---|
| A and B<br>(Mandatory) | 4.50–5.50<br>2.70–3.30 | 3.30–4.50 | V |
| A, B, and C<br>(Optional) | 4.50–5.50<br>2.70–3.30<br>1.62–1.98 | 3.30–4.50<br>1.98–2.70 | V |

While the supply power is maintained within the voltage range defined in Table 7, the $I_{CC}$ current transients shall not exceed the values defined in Table 8.

**Table 8: Transients on ICC Current**

| Class | Maximum charge [a] | Maximum duration | Maximum variation [b] of $I_{CC}$ |
|---|---|---|---|
| A | 20 nA.s | 400 ns | 100 mA |
| B | 10 nA.s | 400 ns | 50 mA |
| C | 6 nA.s | 400 ns | 30 mA |

a The maximum charge is half the product of the maximum duration and the maximum variation.

b The maximum variation is the difference between the peak transient Icc current and the average value of the Icc current over 1 ms measured for the ICC at the time of the transient.

For proprietary reasons terminals may support the capability to negotiate with the ICC the voltage class to be used, but this is outside the scope of EMV, and there is no requirement for ICCs conforming to this specification to support such negotiation. If the ICC returns a class indicator in the ATR as defined in ISO/IEC 7816-3, the ATR may be rejected in an EMV compliant terminal. To avoid interoperability problems, any class indicator used should be returned in the cold ATR; to guarantee that the ICC will be accepted in the event that the cold ATR is rejected, the warm ATR shall be one of the basic ATRs defined in section 8.

**Note:** It is strongly recommended that the current consumption of ICCs is maintained at as low a value as possible, since the maximum current consumption allowable for the ICC may be reduced in future versions of this specification. Issuers of ICCs bearing multisector applications should ensure that the IC used has a current requirement compatible with all terminals (from all sectors) in which the ICC might be used.

### 5.3.7 Contact Resistance

The contact resistance as measured across a pair of clean ICC and clean nominal IFD contacts shall be less than 500 mΩ throughout the design life of an ICC (see ISO/IEC 10373 for test method).

**Note:** A nominal IFD contact may be taken as a minimum of 1.25 µm of gold over 5.00 µm of nickel.

# 5.4 Mechanical Characteristics of the Terminal

This section describes the mechanical characteristics of the terminal interface device.

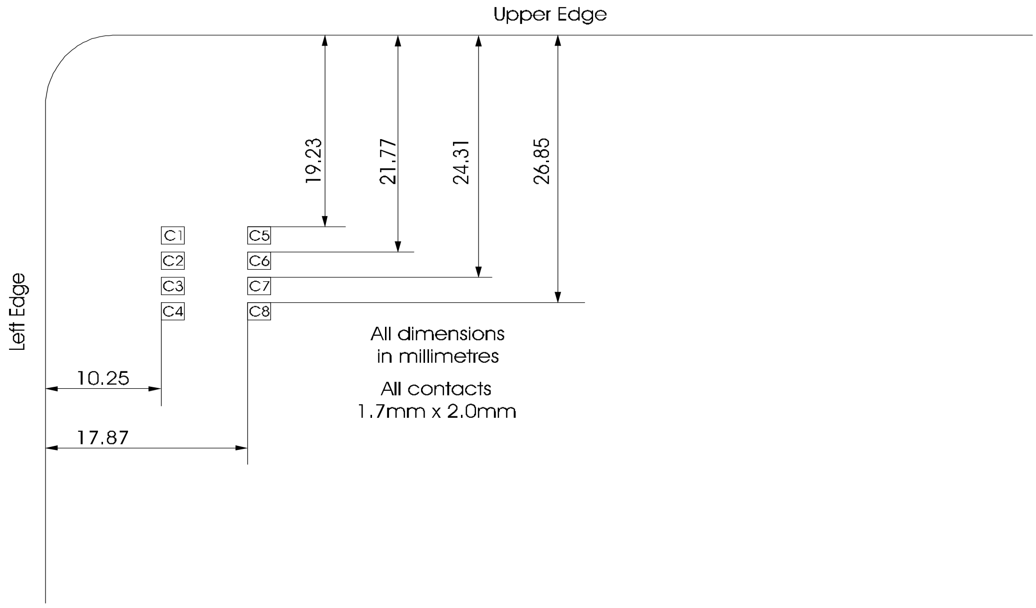
## 5.4.1 Interface Device

The IFD into which the ICC is inserted shall be capable of accepting ICCs having the following characteristics:

- Physical characteristics compliant with ISO/IEC 7816-1
- Contacts on the front, in the position compliant with Figure 2 of ISO/IEC 7816-2
- Embossing compliant with ISO/IEC 7811-1

The IFD contacts shall be located such that if an ICC having contacts with the dimensions and locations specified in Figure 3 is inserted into the IFD, correct connection of all contacts shall be made. The IFD should have no contacts present other than those needed to connect to ICC contacts C1 to C8.

**Figure 3: Terminal Contact Location and Dimensions**




Location guides and clamps (if used) should cause no damage to ICCs, particularly in the areas of the magnetic stripe, signature panel, embossing, and hologram.

**Note:** As a general principle, an ICC should be accessible to the cardholder at all times. Where the ICC is drawn into the IFD, a mechanism should exist to return the ICC to the cardholder in the event of a failure (for example, loss of power).

### 5.4.2 Contact Forces

The force exerted by any one IFD contact on the corresponding ICC contact shall be in the range 0.2 N to 0.6 N.

### 5.4.3 Contact Assignment

The assignment of the IFD contacts shall be as shown in Table 9:

**Table 9: IFD Contact Assignment**

| | | | |
|---|---|---|---|
| C1 | VCC | C5 | GND |
| C2 | RST | C6 | Not used for class A<br>RFU for classes B and C |
| C3 | CLK | C7 | I/O |
| C4 | Not used; need not be physically present | C8 | Not used; need not be physically present |

# 5.5 Electrical Characteristics of the Terminal

This section describes the electrical characteristics of the signals as measured at the IFD contacts.

## 5.5.1 Measurement Conventions

All measurements are made at the point of contact between the ICC and the IFD contacts and are defined with respect to GND contact over an ambient temperature range 5° C to 40° C unless otherwise specified by the manufacturer. The internal temperature of the terminal should be limited to avoid damage to ICCs.

All currents flowing out of the terminal are considered positive.

## 5.5.2 Input/Output (I/O)

This contact is used as an output (transmission mode) to transmit data to the ICC or as an input (reception mode) to receive data from the ICC. During operation, the terminal and the ICC should not both be in transmission mode. In the event that this condition occurs, the state (voltage level) of the contact is indeterminate and no damage shall occur to the terminal.

When both the terminal and the ICC are in reception mode, the contact shall be in the high state. The terminal shall not pull I/O high unless VCC is powered and stable within the tolerances specified in section 5.5.6. See the contact activation sequence specified in section 6.1.2.

The terminal shall limit the current flowing into or out of the I/O contact to ±15 mA at all times.

### 5.5.2.1 Transmission Mode

When in transmission mode, the terminal shall send data to the ICC with the characteristics shown in Table 10:

**Table 10: Electrical Characteristics of I/O for Terminal Transmission**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{OH}$ | $0 < I_{OH} < 20$ μA | $0.8 \times V_{CC}$ | $V_{CC}$ | V |
| $V_{OL}$ | $-0.5$ mA $< I_{OL} < 0$ | 0 | $0.15 \times V_{CC}$ | V |
| $t_R$ and $t_F$ | $C_{IN(ICC)} = 30$ pF max. | — | 0.8 | μs |
| Signal perturbations | Signal low | – 0.25 | $0.15 \times V_{CC}$ | V |
| | Signal high | $0.8 \times V_{CC}$ | $V_{CC} + 0.25$ | V |

Unless transmitting, the terminal shall set its I/O line driver to reception mode.

#### 5.5.2.2 Reception Mode

When in reception mode, the terminal shall correctly interpret signals from the ICC having the characteristics shown in Table 11:

**Table 11: Electrical Characteristics of I/O for Terminal Reception**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{IH}$ | | 0.6 x $V_{CC}$ | $V_{CC}$ | V |
| $V_{IL}$ | | 0 | 0.20 x $V_{CC}$ | V |
| $t_R$ and $t_F$ | | — | 1.2 | µs |

### 5.5.3 Contact C6

C6 shall be electrically isolated. Electrically isolated means that the resistance measured between C6 and any other contact shall be ≥10MΩ with an applied voltage of 5V DC. If connected in existing class A terminals, C6 shall be maintained at a potential between GND and 1.05 x $V_{CC}$ throughout the card session.

**Note:** Keeping C6 isolated in new class A terminals facilitates its use for other purposes if so defined in future versions of this specification.

### 5.5.4 Clock (CLK)

The terminal shall generate a CLK signal having the characteristics shown in Table 12:

**Table 12: Electrical Characteristics of CLK from Terminal**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{OH}$ | $0 < I_{OH} < 50$ µA | 0.8 x $V_{CC}$ | $V_{CC}$ | V |
| $V_{OL}$ | – 50 µA < $I_{OL}$ < 0 | 0 | 0.15 x $V_{CC}$ | V |
| $t_R$ and $t_F$ | $C_{IN(ICC)}$ = 30 pF max. | — | 8% of clock period | |
| Signal perturbations | Signal low | – 0.25 | 0.15 x $V_{CC}$ | V |
| | Signal high | 0.8 x $V_{CC}$ | $V_{CC}$ + 0.25 | V |

Duty cycle shall be between 45% and 55% of the period during stable operation.

The terminals subject to new approval shall use a clock signal frequency signal approaching the maximum of 5 MHz.

The clock signal frequency (CLK) shall be as shown in Table 13:

**Table 13: Clock Frequency (CLK) Requirements**

| Minimum | Maximum | Unit | Date for application |
|---|---|---|---|
| 2.8 | 5 | MHz | Approved terminals due for renewal: to end December 2025 |
| 4.7 | 5 | MHz | Approved terminals due for renewal: from January 2026 |
| 4.7 | 5 | MHz | New terminal approval |

Frequency shall not change by more than ± 1% throughout answer to reset and the following stages of a card session (see section 6).

### 5.5.5 Reset (RST)

The terminal shall generate a RST signal having the characteristics shown in Table 14:

**Table 14: Electrical Characteristics of RST from Terminal**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{OH}$ | $0 < I_{OH} < 50$ μA | 0.8 x $V_{CC}$ | $V_{CC}$ | V |
| $V_{OL}$ | – 50 μA < $I_{OL}$ < 0 | 0 | 0.15 x $V_{CC}$ | V |
| $t_R$ and $t_F$ | $C_{IN(ICC)}$ = 30 pF max. | — | 0.8 | μs |
| Signal perturba-tions | Signal low | – 0.25 | 0.15 x $V_{CC}$ | V |
| | Signal high | 0.8 x $V_{CC}$ | $V_{CC}$ + 0.25 | V |

### 5.5.6 Supply Voltage (VCC)

The terminal shall generate a $V_{CC}$ within one of the range(s) specified in Table 15 below for the class(es) supported, and shall be capable of delivering the corresponding steady state output current whilst maintaining $V_{CC}$ within that range. If the terminal supports more than one class, it shall always generate a $V_{CC}$ from the class containing the highest voltage range available.

For proprietary reasons terminals may support the capability to negotiate with the ICC the voltage class to be used, but this is outside the scope of EMV, and is not supported by ICCs conforming to this specification. Attempting class negotiation with such an ICC may result in the ICC being rejected.

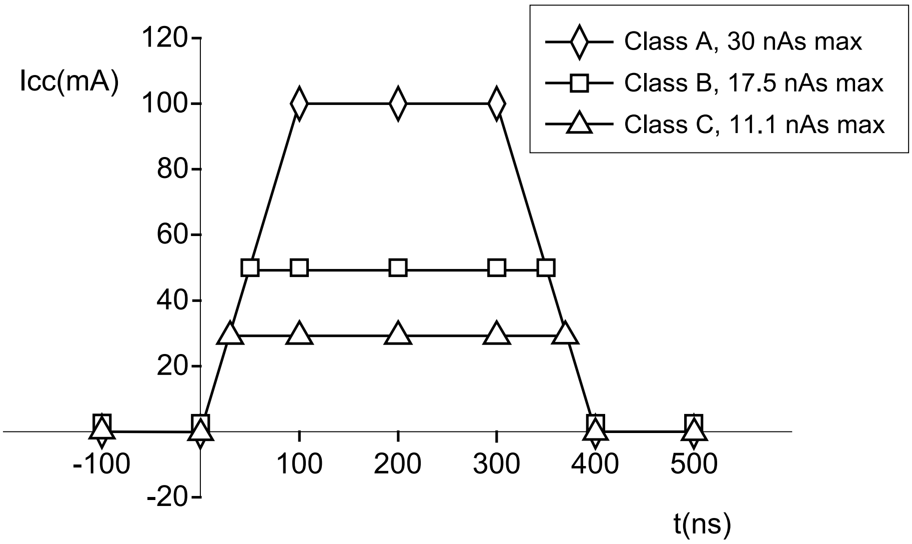
The supply shall be protected from transients and surges caused by internal operation of the terminal and from external interference introduced via power leads, communications links, etc. $V_{CC}$ shall never be less than –0.25V with respect to ground.

**Table 15: Terminal Supply Voltage and Current**

| Symbol | Conditions | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Class A<br>Class B<br>Class C | 4.60<br>2.76<br>1.66 | 5.40<br>3.24<br>1.94 | V |
| $I_{CC}$ | Class A<br>Class B<br>Class C | 55<br>55<br>35 | | mA |

During normal operation of an ICC, current pulses cause voltage transients on VCC as measured at the ICC contacts. The power supply shall be able to counteract transients in the current consumption of the ICC having characteristics within the maximum charge envelope applicable to the class of operation as shown in Figure 4, ensuring that VCC remains within the range specified.

**Figure 4: Maximum Current Pulse Envelopes**




**Note:** Terminals may be designed to be capable of delivering more than required current, but it is recommended that terminals limit the steady state current that can be delivered to a maximum of 200 mA.

## 5.5.7 Contact Resistance

The contact resistance as measured across a pair of clean IFD and clean nominal ICC contacts shall be less than 500 mΩ throughout the design life of a terminal (see ISO/IEC 7816-1 for test method).

**Note:** A nominal ICC contact may be taken as 1.25 µm of gold over 5.00 µm of nickel.

## 5.5.8 Short Circuit Resilience

The terminal shall not be damaged in the event of fault conditions such as a short circuit between any combinations of contacts. The terminal shall be capable of sustaining a short circuit of any duration between any or all contacts without suffering damage or malfunction, for example, if a metal plate is inserted.

## 5.5.9 Powering and Depowering of Terminal with ICC in Place

If the terminal is powered on or off with an ICC in place, all signal voltages shall remain within the limits specified in section 5.5, and contact activation and deactivation sequences and timings, as described in sections 6.1.2 and 6.1.5 respectively, shall be respected.

# 6 Card Session

This section describes all stages involved in a card session from insertion of the ICC into the IFD through the execution of the transaction to the removal of the ICC from the IFD.

## 6.1 Normal Card Session

This section describes the processes involved in the execution of a normal transaction.

### 6.1.1 Stages of a Card Session

A card session is comprised of the following stages:

1. Insertion of the ICC into the IFD and connection and activation of the contacts.
2. Reset of the ICC and establishment of communication between the terminal and the ICC.
3. Execution of the transaction(s).
4. Deactivation of the contacts and removal of the ICC.

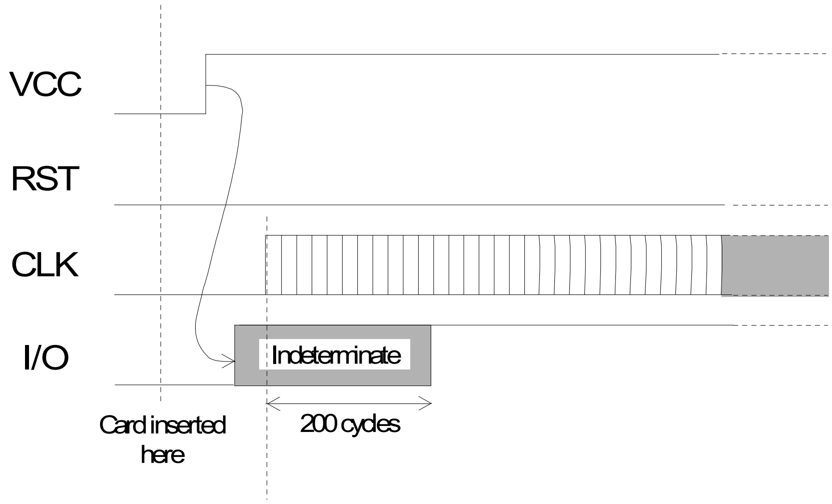
### 6.1.2 ICC Insertion and Contact Activation Sequence

On insertion of the ICC into the IFD, the terminal shall ensure that all signal contacts are in state L with values of $V_{OL}$ as defined in section 5.5 and that $V_{CC}$ is 0.4 V or less at the instant galvanic contact is made. When the ICC is correctly seated within the IFD, the contacts shall be activated as follows (see Figure 5):

- RST shall be maintained by the terminal in state L throughout the activation sequence.
- Following establishment of galvanic contact but prior to activation of I/O or CLK, VCC shall be powered.
- Following verification by the terminal that $V_{CC}$ is stable and within the limits defined in section 5.5.6, the terminal shall set its I/O line driver to reception mode and shall provide CLK with a suitable and stable clock as defined in section 5.5.4. The I/O line driver in the terminal may be set to reception mode prior to application of the clock but shall be set to reception mode no later than 200 clock cycles after application of the clock.

**Note:** The terminal may verify the state of $V_{CC}$ by measurement, by waiting sufficient time for it to stabilise according to the design of the terminal, or otherwise. The state of the I/O line after the terminal has set its I/O line driver to reception mode is dependent upon the state of the I/O line driver in the ICC (see section 6.1.3.1).

**Figure 5: Contact Activation Sequence**



### 6.1.3 ICC Reset

The ICC shall answer to reset asynchronously using active low reset.

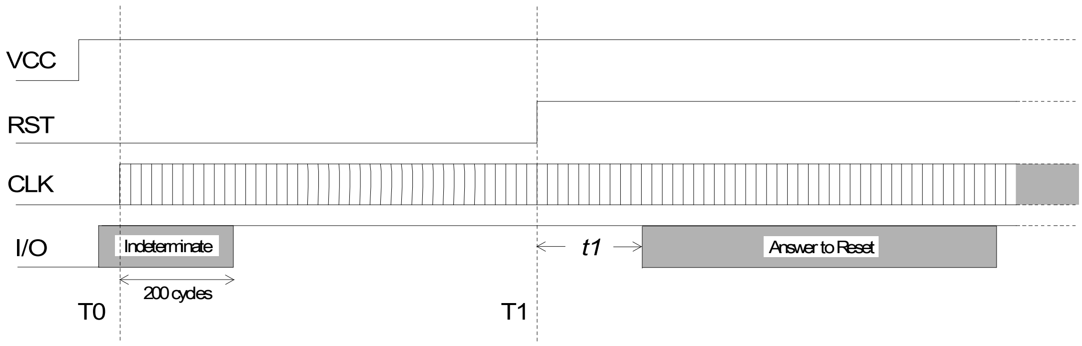
The means of transportation of the answer to reset (ATR) are described in section 7 and its contents are described in sections 8.2 and 8.3.

#### 6.1.3.1 Cold Reset

Following activation of the contacts according to section 6.1.2, the terminal shall initiate a cold reset and obtain an ATR from the ICC as follows (see Figure 6):

- The terminal shall apply CLK at a notional time T0.
- Within a maximum of 200 clock cycles following T0, the ICC shall set its I/O line driver to reception mode. Since the terminal shall also have set its I/O line driver to reception mode within this period, the I/O line is guaranteed to be in state H no later than 200 clock cycles following time T0.
- The terminal shall maintain RST in state L through time T0 and for a period of between 40,000 and 45,000 clock cycles following time T0 to time T1, when it shall set RST to state H.
- The answer to reset on I/O from the ICC shall begin between 400 and 40,000 clock cycles after time T1 (time *t1* in Figure 6).
- The terminal shall have a reception window which is opened no later than 380 clock cycles after time T1 and closed no earlier than 42,000 clock cycles after time T1 (time *t1* in Figure 6). If no answer to reset is received from the ICC, the terminal shall initiate the deactivation sequence no earlier than 42,001 clock cycles after time T1, and no later than 42,000 clock cycles plus 50ms after time T1.

**Figure 6: Cold Reset Sequence**

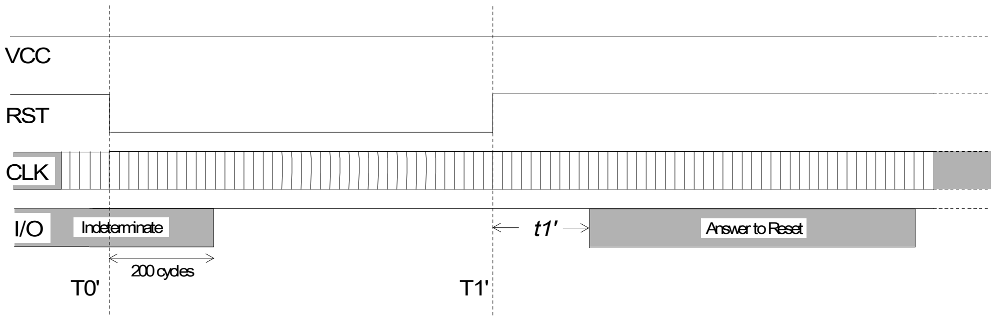


### 6.1.3.2 Warm Reset

If the ATR received following a cold reset as described in section 6.1.3.1 does not conform to the specification in section 8, the terminal shall initiate a warm reset and obtain an ATR from the ICC as follows (see Figure 7):

- A warm reset shall start at a notional time T0', at which time the terminal shall set RST to state L.
- The terminal shall maintain VCC and CLK stable and within the limits defined in sections 5.5.4 and 5.5.6 throughout the warm reset sequence.
- Within a maximum of 200 clock cycles following T0', the ICC and terminal shall set their I/O line drivers to reception mode. The I/O line therefore is guaranteed to be in state H no later than 200 clock cycles following time T0'.
- The terminal shall maintain RST in state L from time T0' for a period of between 40,000 and 45,000 clock cycles following time T0' to time T1', when it shall set RST to state H.
- The answer to reset on I/O from the ICC shall begin between 400 and 40,000 clock cycles after time T1' (time *t1'* in Figure 7).
- The terminal shall have a reception window which is opened no later than 380 clock cycles after time T1' and closed no earlier than 42,000 clock cycles after time T1' (time *t1'* in Figure 7). If no answer to reset is received from the ICC, the terminal shall initiate the deactivation sequence no earlier than 42,001 clock cycles after time T1', and no later than 42,000 clock cycles plus 50ms after time T1'.

**Figure 7: Warm Reset Sequence**




**Note:** Figure 7 indicates that the terminal may initiate the warm reset sequence during the time that the card is still transmitting the cold ATR, and in the event that it does, the card shall be able to respond correctly with the warm ATR.

Besides following a rejected ATR or an erroneous PPS exchange, if the terminal initiates a warm reset sequence during the later stages of the transaction, the card shall respond correctly with the warm ATR.

### 6.1.4 Execution of a Transaction

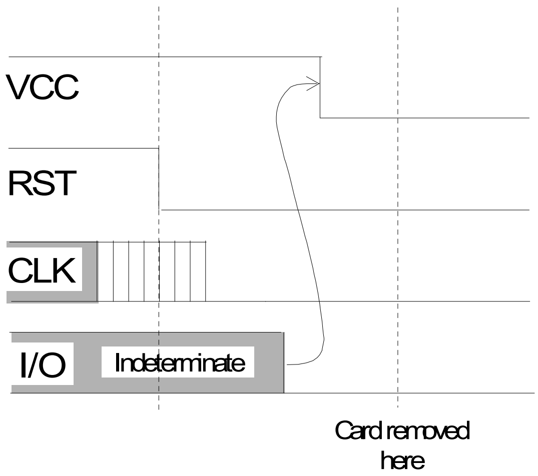
Selection of the application in the ICC and the subsequent exchange of information between the ICC and the terminal necessary to perform a transaction are described in Level 2 Specifications (See General Bulletin GB No 49).

### 6.1.5 Contact Deactivation Sequence

As the final step in the card session, upon normal or abnormal termination of the transaction (including withdrawal of the ICC from the IFD during a card session), the terminal shall deactivate the IFD contacts as follows (see Figure 8):

- The terminal shall initiate the deactivation sequence by setting RST to state L.
- Following the setting of RST to state L but prior to depowering VCC, the terminal shall set CLK and I/O to state L.
- Following the setting of RST, CLK, and I/O to state L but prior to galvanic disconnection of the IFD contacts, the terminal shall depower VCC. $V_{CC}$ shall be 0.4 V or less prior to galvanic disconnection of the IFD contacts.
- The deactivation sequence shall be completed within 100 ms. This period is measured from the time that RST is set to state L to the time that $V_{CC}$ reaches 0.4 V or less.

**Figure 8: Contact Deactivation Sequence**



## 6.2 Abnormal Termination of Transaction Process

If an ICC is prematurely removed from a terminal during execution of a transaction at speeds of up to 1 m/s, the terminal shall be capable of sensing the movement of the ICC relative to the IFD contacts, and of deactivating all IFD contacts in the manner described in section 6.1.5 before the relative movement exceeds 1 mm. No electrical or mechanical damage shall be caused to the ICC under these conditions.

**Note:** For ‘sliding carriage’ type IFDs, it may be possible for the terminal to sense the movement of the ICC/IFD contact sub-assembly relative to the main body of the IFD. In this event, it is not mandatory to be able to sense the movement of the ICC relative to the IFD contacts, but deactivation of the contacts shall be complete before any electrical contact is broken between the ICC and IFD.

# 7 Physical Transportation of Characters

During the transaction process, data is passed bi-directionally between the terminal and the ICC over the I/O line in an asynchronous half duplex manner. A clock signal is provided to the ICC by the terminal, and this shall be used to control the timing of this exchange. The mechanism of exchanging bits and characters is described below. It applies during the answer to reset and is also used by both transmission protocols as described in section 9.

## 7.1 Bit Duration

The bit duration used on the I/O line is defined as an elementary time unit (etu). A linear relationship exists between the etu on the I/O line and CLK frequency (*f*).

During the answer to reset, the bit duration is known as the initial etu, and is given by the following equation:

$$\text{initial etu} = \frac{372}{f} \text{ seconds, where } f \text{ is in Hertz}$$

Following the answer to reset (and establishment of the global parameters F and D, as described in section 8), the bit duration is known as the current etu, and is given by the following equation:

$$\text{current etu} = \frac{\mathrm{F}}{\mathrm{D}f} \text{ seconds, where } f \text{ is in Hertz}$$

**Note:** For the basic answer(s) to reset described in this specification, only values of F = 372 and D = 1 are supported. In the following sections, “etu” indicates current etu unless otherwise specified.

## 7.2 Character Frame

Data is passed over the I/O line in a character frame as described below. The convention used is specified in the initial character (TS) transmitted by the ICC in the ATR (see section 8.3.1).

Prior to transmission of a character, the I/O line shall be in state H.

A character consists of 10 consecutive bits (see Figure 9):

- 1 start bit in state L
- 8 bits, which comprise the data byte
- 1 even parity checking bit

The start bit is detected by the receiving end by periodically sampling the I/O line. The sampling time should be less than or equal to 0.2 etu.

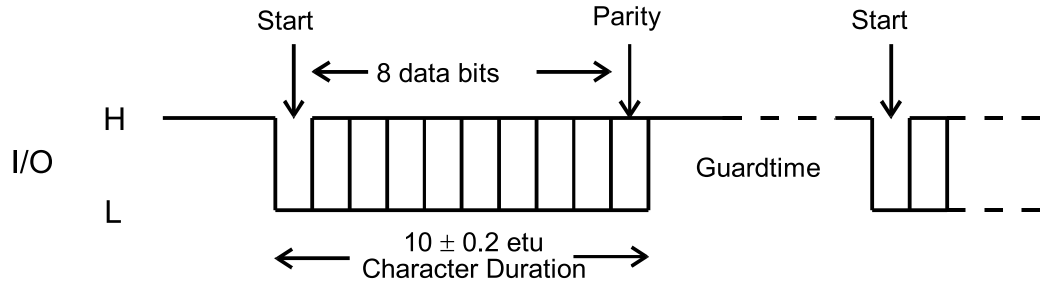
The number of logic ones in a character shall be even. The 8 bits of data and the parity bit itself are included in this check but the start bit is not.

The time origin is fixed as midway between the last observation of state H and the first observation of state L. The existence of a start bit should be verified within 0.7 etu. Subsequent bits should be received at intervals of (n + 0.5 ± 0.2) etu (n being the rank of the bit). The start bit is bit 1.

Within a character, the time from the leading edge of the start bit to the trailing edge of the nth bit is (n ± 0.2) etu.

The interval between the leading edges of the start bits of two consecutive characters is comprised of the character duration (10 ± 0.2) etu, plus a guardtime. Under error free transmission, during the guardtime both the ICC and the terminal shall be in reception mode (I/O line in state H). For T=0 only, if the ICC or terminal as receiver detects a parity error in a character just received, it shall set I/O to state L to indicate the error to the sender (see section 9.2.3).

**Figure 9: Character Frame**




At the terminal transport layer (TTL), data shall always be passed over the I/O line most significant (m.s.) byte first. The order of bits within a byte (that is, whether the least significant (l.s.) or m.s. bit is transferred first) is specified in character TS returned in the answer to reset (see section 8.3).

# 8 Answer to Reset

After being reset by the terminal as described in section 6.1.3, the ICC answers with a string of bytes known as the ATR. These bytes convey information to the terminal that defines certain characteristics of the communication to be established between the ICC and the terminal. The method of transporting these bytes, and their meaning, is described below.

**Note:** In sections 8 and 9, the m.s. bit of a character refers to bit b8 and the l.s. bit refers to bit b1. A value in straight single quotes is coded in hexadecimal notation; for example, 'A' or '3F'.

## 8.1 Physical Transportation of Characters Returned at Answer to Reset

This section describes the structure and timing of the characters returned at the answer to reset.

The bit duration is defined in section 7.1, and the character frame is defined in section 7.2.

During the answer to reset, the minimum interval between the leading edges of the start bits of two consecutive characters shall be 12 initial etus, and the maximum interval between the leading edges of the start bits of two consecutive characters shall be 480 initial etus.

The ICC shall transmit all the characters to be returned during an answer to reset (warm or cold) within 960 initial etus.[2] This time is measured between the leading edge of the start bit of the first character (TS) and 12 initial etus after the leading edge of the start bit of the last character.

[2] The maximum time allowed for the answer to reset varies according to clock frequency, since the period represented by an etu is frequency dependent (see section 7.1).

## 8.2 Characters Returned by ICC at Answer to Reset

The number and coding of the characters returned by the ICC at the answer to reset varies depending upon the transmission protocol(s) and the values of the transmission control parameters supported. This section describes two basic answers to reset, one for ICCs supporting T=0 only and the other for ICCs supporting T=1 only. It defines the characters to be returned and the allowable ranges of values for the transmission control parameters. ICCs returning one of the two answers to reset described here are assured correct operation and interoperability in terminals conforming to this specification.

For proprietary reasons ICCs may optionally support more than one transmission protocol, but one of the supported protocols shall be T=0 or T=1. The first offered protocol shall be T=0 or T=1, and the terminal shall continue the card session using the first offered protocol unless for proprietary reasons it supports a mechanism for selecting an alternative protocol offered by the ICC. Support for such a mechanism is not required by, and is beyond the scope of these specifications.

**Note:** This specification does not support ICCs having both T=0 and T=1 protocols present at the same time. This can only be achieved by proprietary means beyond the scope of this specification.

**Note:** It is strongly recommended that T=1 protocol should be used for optimum performance and since T=0 protocol may not be supported in future versions of this specification.

Also for proprietary reasons ICCs may optionally support other values of the transmission control parameters at the issuer's discretion. However, such support is considered outside the scope of this specification and such ICCs may be rejected at terminals conforming to this specification, which need not have the corresponding additional proprietary functionality required to support the ICC.

The characters returned by the ICC at the answer to reset for the basic answers to reset are shown in Table 16 and Table 17. The characters are shown in the order in which they are sent by the ICC, that is, TS first.

If protocol type T=0 only is supported (character-oriented asynchronous transmission protocol), the characters returned shall be as shown in Table 16:

**Table 16: Basic ATR for T=0 Only**

| Character | Value | Remarks |
|---|---|---|
| TS | '3B' or '3F' | Indicates direct or inverse convention |
| T0 | '6x' for a cold ATR<br><br>'6x' or '4x' for a warm ATR | TB1 and TC1 present; x indicates the number of historical bytes present<br><br>TB1 optionally present and TC1 present; x indicates the number of historical bytes present |
| TB1 | '00' | The presence of TB1 has been deprecated by ISO/IEC 7816-3:2006, however the value '00' is kept here for backwards compatibility with existing terminals and its presence is mandatory for a basic cold ATR but is optional for a basic warm ATR |
| TC1 | '00' or 'FF' | Indicates the amount of extra guardtime required. Value 'FF' has a special meaning (see section 8.3.3.3) |

If protocol type T=1 only is supported (block-oriented asynchronous transmission protocol), the characters returned shall be as shown in Table 17:

**Table 17: Basic ATR for T=1 Only**

| Character | Value | Remarks |
|---|---|---|
| TS | '3B' or '3F' | Indicates direct or inverse convention |
| T0 | 'Ex' for a cold ATR<br><br>'Ex' or 'Cx' for a warm ATR | TB1 to TD1 present; x indicates the number of historical bytes present<br><br>TB1 optionally present and TC1, TD1 present; x indicates the number of historical bytes present |
| TB1 | '00' | The presence of TB1 has been deprecated by ISO/IEC 7816-3:2006, however the value '00' is kept here for backwards compatibility with existing terminals and its presence is mandatory for a basic cold ATR but is optional for a basic warm ATR |
| TC1 | '00' or 'FF' | Indicates amount of extra guardtime required. Value 'FF' has special meaning (see section 8.3.3.3) |
| TD1 | '81' | TA2, TB2, and TC2 absent; TD2 present; T=1 to be used |
| TD2 | '31' | TA3 and TB3 present; TC3 and TD3 absent; T=1 to be used |
| TA3 | '7C' to 'FE' | Returns IFSI, which indicates initial value for information field size for the ICC and IFSC of 124–254 bytes |
| TB3 | m.s. nibble '0' to '4'<br>l.s. nibble '0' or '1' | BWI = 0 to 4<br>CWI = 0 or 1 |
| TCK | See section 8.3.4 | Check character |

## 8.3 Character Definitions

This section provides detailed descriptions of the characters that may be returned at the answer to reset.

Each character description includes the following information:

- title
- explanation of usage as described in ISO/IEC 7816-3
- basic response (This response shall always be used in a warm ATR to ensure interoperability.)
- required terminal behaviour in the event that a terminal receives characters outside the range allowed by EMV

The ‘basic response’ indicates the presence or absence of the character, and the allowable range of values it may take (if present) if it is to conform to one of the basic ATRs. The description of a basic response (even though indicated by ‘shall’) is not intended to preclude the use of other values of the characters, nor the omission/inclusion of a character at the issuer’s discretion. For example, the ICC may return additional characters if it supports more than one transmission protocol (see section 9). However, only ICCs returning a basic ATR, or an ATR supported by the minimum required terminal functionality described below, are guaranteed to be supported correctly in interchange. The warm ATR shall always be basic.

Terminals conforming to this specification are required to support the basic ATRs described here together with the additional requirements specified in ‘terminal behaviour’. Terminals may thus reject an ATR containing interface bytes not described in, or having values not specified in, this specification. However, terminals may correctly interpret such an ATR if it is returned by an ICC for proprietary (for example, national) use. Such terminal functionality is not mandatory and is beyond the scope of this specification. As a general principle, a terminal should accept a non-basic ATR if it is able to function correctly with it.

Terminals shall be capable of checking the parity of characters returned in the answer to reset, but not necessarily as they are received. If the terminal detects a parity error, it shall reject the ICC.

Table 18 describes the action indicated by several terms in the following character descriptions:

**Table 18: Terminal Behaviour**

| If it is indicated that a terminal shall: | then: |
|---|---|
| reject an ATR | • If the terminal is rejecting a cold ATR, the terminal shall issue a warm reset.<br>• If the terminal is rejecting a warm ATR, the terminal shall terminate the card session by deactivating the ICC contacts. |
| reject an ICC | The terminal shall terminate the card session by deactivating the ICC contacts. |
| accept an ATR | The terminal shall accept the ATR, but *only* if the requirements specified in this section for all other characters are also met. |

### 8.3.1 TS - Initial Character

TS performs two functions. It provides a known bit pattern to the terminal to facilitate bit synchronisation, and it indicates the logic convention to be used for the interpretation of the subsequent characters.

Using inverse logic convention, a low state L on the I/O line is equivalent to a logic one, and the m.s. bit of the data byte is the first bit sent after the start bit. Using direct logic convention, a high state H on the I/O line is equivalent to a logic one, and the l.s. bit of the data byte is the first bit sent after the start bit. The first four bits LHHL are used for bit synchronisation.

*Basic responses:* The ICC shall return an ATR containing TS having one of two values:

- `(H)LHHLLLLLLH` inverse convention value '3F'
- `(H)LHHLHHHLLH` direct convention value '3B'

The convention used may differ between cold and warm resets.

*Terminal behaviour:* The terminal shall be capable of supporting both inverse and direct convention and shall accept an ATR containing TS having a value of either '3B' or '3F'. An ICC returning an ATR containing TS having any other value shall be rejected.

**Note:** It is strongly recommended that a value of '3B' is returned by the ICC since a value of '3F' may not be supported in future versions of this specification.

### 8.3.2 T0 - Format Character

T0 is comprised of two parts. The m.s. nibble (b5–b8) is used to indicate whether the subsequent characters TA1 to TD1 are present. Bits b5–b8 are set to the logic one state to indicate the presence of TA1 to TD1, respectively. The l.s. nibble (b1–b4) indicates the number of optional historical bytes present (0 to 15). (See Table 19 for the basic response coding of character T0.)

*Basic responses:* If T=0 only is to be used, a cold basic ATR shall contain T0 = '6x', indicating that characters TB1 and TC1 are present and a warm basic ATR shall contain T0 = '6x' or '4x', indicating that optional character TB1 is present or not (respectively) and character TC1 is present. If T=1 only is to be used, a cold basic ATR shall contain T0 = 'Ex', indicating that characters TB1 to TD1 are present and a warm basic ATR shall contain T0 = 'Ex' or 'Cx' indicating that optional character TB1 is present or not (respectively) and characters TC1, TD1 are present. The value of 'x' represents the number of optional historical bytes to be returned.[3]

*Terminal behaviour:* The terminal shall accept an ATR containing T0 of any value provided that the value returned correctly indicates and is consistent with the interface characters TA1 to TD1 and historical bytes actually returned.

**Table 19: Basic Response Coding of Character T0**

| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|---|
| T=0 only | 0 | 1 | 1 for a cold ATR<br>0 or 1 for a warm ATR | 0 | x | x | x | x |
| T=1 only | 1 | 1 | 1 for a cold ATR<br>0 or 1 for a warm ATR | 0 | x | x | x | x |

[3] Although their use is not forbidden, the EMV Specifications do not explicitly support the use of historical bytes, and their usage, structure and meaning are outside the scope of EMV. However, if used, it is strongly recommended that they are encoded to have a structure and meaning according to ISO/IEC 7816-4. EMV-compliant terminals should ignore any historical bytes present in the ATR. However, if an EMV-compliant terminal does support historical bytes, it should never be designed in such a way that non-recognition or misinterpretation of any historical bytes present in the ATR causes termination of the transaction. Since Issuers are free to encode the historical bytes in any way they choose, they should recognise that unintentional conflict of coding between cards may exist, leading to misinterpretation at terminals. Great care should be exercised by the terminal that it is able to unambiguously identify a card before interpreting any historical bytes returned in the ATR.

### 8.3.3 TA1 to TC3 - Interface Characters

TA1 to TC3 convey information that shall be used during exchanges between the terminal and the ICC subsequent to the answer to reset. They indicate the values of the transmission control parameters F, D, and N, and the IFSC, block waiting time integer (BWI), and character waiting time integer (CWI) applicable to T=1 as defined in ISO/IEC 7816-3. The information contained in TA1, TC1, and TA2 shall apply to all subsequent exchanges irrespective of the protocol type to be used.

#### 8.3.3.1 TA1

TA1 conveys the values of FI and DI where:

- the m.s. nibble FI is used to determine the value of F, the clock rate conversion factor, which may be used to modify the frequency of the clock provided by the terminal subsequent to the answer to reset
- the l.s. nibble DI is used to determine the value of D, the bit rate adjustment factor, which may be used to adjust the bit duration used subsequent to the answer to reset

See section 7.1 for calculation of the bit duration subsequent to the answer to reset (current etu).

Default values of FI = 1 and DI = 1 indicating values of F = 372 and D = 1, respectively, shall be used during the answer to reset.

The cold ATR shall contain TA1 as below:

- If TA2 is returned with b5 = 0 (specific mode, parameters defined by the interface bytes), TA1 shall be coded with '13' indicating the values of F = 372 and D = 4, respectively. In such case the warm ATR shall be a basic ATR.
- If TA2 is not returned (negotiable mode), TA1 shall be coded with a most significant nibble (m.s. nibble) different from 0 and therefore indicate a maximum frequency of at least 5 MHz. TA1 shall be coded with a least significant nibble (l.s. nibble) greater than or equal to 3 and therefore indicating a value of D greater than or equal to 4. In such case the warm ATR shall be a basic ATR.

ICCs returning a cold or warm ATR different than explained above are not compliant with this specification.

*Basic response:* The ATR shall not contain TA1 and thus the default values of F = 372 and D = 1 shall continue be used during all subsequent exchanges.

*Terminal behaviour:* If TA1 is present in the ATR (indicated by b5 of T0 set to 1) and TA2 is returned with b5 = 0 (specific mode, parameters defined by the interface bytes), the terminal shall:

- Accept the ATR if the value of TA1 is in the range '11' to '13',[4] and immediately implement the values of F and D indicated (F = 372 and D = 1, 2, or 4).
- Reject the ATR if the value of TA1 is not in the range '11' to '13', unless it is able to support and immediately implement the conditions indicated.

If TA1 is present in the ATR (indicated by b5 of T0 set to 1) and TA2 is not returned (negotiable mode), the terminal shall accept the ATR and shall continue using the default values of D = 1 and F = 372 during all subsequent exchanges, unless it supports a proprietary technique for negotiating the parameters to be used.

If TA1 is absent from the ATR, the default values of D = 1 and F = 372 shall be used during all subsequent exchanges.

#### 8.3.3.2 TB1

The usage of TB1 has been deprecated by ISO/IEC 7816-3:2006 but the value '00' is kept in this version of the specification for backwards compatibility with existing terminals.

If the cold ATR does not contain TB1 = '00' for proprietary reasons the warm ATR shall be a basic ATR, otherwise the ICC is not compliant with this specification.

**Note:** A cold ATR without TB1 = '00' will cause default communication speed with terminals compliant to EMV 4.3 and earlier versions by enforcing them to initiate the warm basic ATR (default, D = 1).

*Basic response*: The cold ATR shall contain TB1 = '00'. The warm ATR may optionally contain TB1 = '00'.

*Terminal behaviour:* In response to a cold reset or a warm reset, the terminal shall accept an ATR containing TB1 with any value or not containing TB1 and continue the card session. No signals shall ever be generated on contact C6 except as described in section 5.5.3 existing class A terminals may maintain C6 at a potential between GND and 1.05 x VCC throughout the card session.

[4] Terminals compliant to version 3.1.1 of the EMV Specifications may reject an ATR (not an ICC) if TA1 is present and coded other than '11'. ATRs indicating the higher allowable values of D will include TA1 coded '12' or '13', and thus may be rejected in 3.1.1 compliant terminals. Therefore, to ensure that an ICC supporting higher data transfer rates is always accepted in 3.1.1 compliant terminals (albeit operating at basic data transfer rates), it is essential that any TA1 indicating higher data rates is present in the cold ATR only, and that a warm ATR is always present which either does not contain TA1, or includes a TA1 having the value '11'.

Date for application:

| | | |
|---|---|---|
| Until January 2026 | Approved terminals due for renewal | may reject a cold ATR which does not contain TB1 = '00' |
| From January 2026 | Approved terminals due for renewal | shall implement the requirements defined in the terminal behaviour above. |

#### 8.3.3.3 TC1

TC1 conveys the value of N, where N is used to indicate the extra guardtime that shall be added to the minimum duration between the leading edges of the start bits of two consecutive characters for subsequent exchanges from the terminal to the ICC. N is binary coded over bits b1–b8 of TC1, and its value represents the number of etus to be added as extra guardtime. TC1='FF' has a special meaning and indicates that the minimum delay between the leading edges of the start bits of two consecutive characters shall be reduced to 12 etus if T=0 is to be used, or 11 etus if T=1 is to be used.

**Note:** TC1 applies only to the timing between two consecutive characters sent from the terminal to the ICC. It does not apply to the timing between consecutive characters sent from the ICC to the terminal, nor does it apply to the timing between two characters sent in opposite directions (see sections 9.2.2.1 and 9.2.4.2.2).

N may take any value between 0 and 255.

If the value of TC1 is in the range '00' to 'FE', between 0 and 254 etus of extra guardtime shall be added to the minimum character to character duration, which for subsequent transmissions shall be between 12 and 266 etus.

If the value of TC1 = 'FF', then the minimum character to character duration for subsequent transmissions shall be 12 etus if T=0 is to be used, or 11 etus if T=1 is to be used.

An ATR containing TC1 having values other than '00' or 'FF' is not compliant with this specification.

*Basic response:* The ATR shall contain TC1 having a value '00' or 'FF'.

*Terminal behaviour:* The terminal shall accept an ATR not containing TC1 (provided that b7 of T0 is set to 0), and shall continue the card session as though TC1 = '00' had been returned.

The basic response coding of character TC1 is shown in Table 20:

**Table 20: Basic Response Coding of Character TC1**

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

#### 8.3.3.4 TD1

TD1 indicates whether any further interface bytes are to be transmitted and information concerning the protocol type(s) where:

- The m.s. nibble is used to indicate whether the characters TA2 to TD2 are present. These bits (b5–b8) are set to the logic one state to indicate the presence of TA2 to TD2 respectively.
- The l.s. nibble provides information concerning the protocol type(s) to be used for subsequent exchanges.

*Basic responses:* The ATR shall not contain TD1 if T=0 only is to be used, and protocol type T=0 shall be used as a default for all subsequent transmissions. The ATR shall contain TD1 = '81' if T=1 only is to be used, indicating that TD2 is present and that protocol type T=1 shall be used for all subsequent transmissions.

**Note:** It is strongly recommended that the ATR contains a TD1 with a value of 'x1' indicating that protocol type T=1 shall be used for all subsequent transmissions for optimum performance and as T=0 protocol may not be supported in future versions of this specification.

*Terminal behaviour:* The terminal shall accept an ATR containing TD1 with the m.s. nibble having any value (provided that the value returned correctly indicates and is consistent with the interface characters TA2 to TD2 actually returned), and the l.s. nibble having a value of '0' or '1'. The terminal shall reject an ATR containing other values of TD1.

The basic response coding of character TD1 is shown in Table 21:

**Table 21: Basic Response Coding of Character TD1**

| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|---|
| T=1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

#### 8.3.3.5 TA2

The presence or absence of TA2 indicates whether the ICC will operate in specific mode or negotiable mode respectively following the answer to reset. When present, TA2 conveys information regarding the specific mode of operation where:

- b8 indicates whether the ICC is capable of changing its mode of operation. It is capable of changing if b8 is set to 0, and unable to change if b8 is set to 1.
- b7–b6 are RFU (set to 00).
- b5 indicates whether the transmission parameters to be used following Answer to Reset are defined in the interface characters or are implicitly known by the terminal. The transmission parameters are defined by the interface characters if b5 is set to 0, or are implicitly known by the terminal if b5 is set to 1.
- l.s. nibble b4-b1 indicates the protocol to be used in specific mode.

*Basic response:* The ATR shall not contain TA2; the absence of TA2 indicates the negotiable mode of operation.

*Terminal behaviour:* The terminal shall accept an ATR containing TA2 provided that all the following conditions are met:

- The protocol indicated in the l.s. nibble is also the first indicated protocol in the ATR.
- b5 = 0
- The terminal is able to support the exact conditions indicated in the applicable interface characters and immediately uses those conditions.

Otherwise, the terminal shall reject an ATR containing TA2.

#### 8.3.3.6 TB2

The usage of TB2 has been deprecated by ISO/IEC 7816-3:2006.

An ATR containing TB2 is not compliant with this specification.

*Basic response:* The ATR shall not contain TB2.

*Terminal behaviour:* The terminal shall accept an ATR containing TB2 and shall ignore any value of TB2 that is returned and continue the card session.

Date for application:

| | | |
|---|---|---|
| Until January 2026 | Approved terminals due for renewal | may reject an ATR containing TB2 |
| From January 2026 | Approved terminals due for renewal | shall implement the requirements defined in the terminal behaviour above. |

#### 8.3.3.7 TC2

TC2 is specific to protocol type T=0 and conveys the work waiting time integer (WI) that is used to determine the maximum interval between the leading edge of the start bit of any character sent by the ICC and the leading edge of the start bit of the previous character sent either by the ICC or the terminal (the work waiting time). The work waiting time is given by 960 x D x WI. See section 9.2.2.1 for additional requirements that are imposed for maximum interval limitations related to the work waiting time and inter-character timing during transmission.

*Basic response:* The ATR shall not contain TC2 and a default value of WI = 10 shall be used during subsequent communication.

*Terminal behaviour:* The terminal shall:

- reject an ATR containing TC2 = '00'
- accept an ATR containing TC2 = '0A'
- reject an ATR containing TC2 having any other value unless it is able to support it.

#### 8.3.3.8 TD2

TD2 indicates whether any further interface bytes are to be transmitted and the protocol type to be used for subsequent transmissions, where:

- The m.s. nibble is used to indicate whether the characters TA3 to TD3 are present. These bits (b5–b8) are set to the logic one state to indicate the presence of TA3 to TD3, respectively.
- The l.s. nibble indicates the protocol type to be used for subsequent exchanges. It shall take the value '1' as T=1 is to be used.

*Basic responses:* The ATR shall not contain TD2 if T=0 is to be used, and the protocol type T=0 shall be used as a default for all subsequent transmissions. The ATR shall contain TD2 = '31' if T=1 is to be used, indicating that TA3 and TB3 are present and that protocol type T=1 shall be used for all subsequent transmissions.

*Terminal behaviour:* The terminal shall accept an ATR containing TD2 with the m.s. nibble having any value (provided that the value returned correctly indicates and is consistent with the interface characters TA3 to TD3 actually returned), and the l.s. nibble having a value of '1' (or 'E' if the l.s. nibble of TD1 is '0'). The terminal shall reject an ATR containing other values of TD2.

The basic response coding of character TD2 is shown in Table 22:

**Table 22: Basic Response Coding of Character TD2**

| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|---|
| T=1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

#### 8.3.3.9 TA3

TA3 (if T=1 is indicated in TD2) returns the information field size integer for the ICC (IFSI), which determines the IFSC, and specifies the maximum length of the information field (INF) of blocks that can be received by the card. It represents the length of IFSC in bytes and may take any value between '7C' and 'FE'. Values of '00' and 'FF' are reserved for future use.

A T=1 ATR not having TA3 or having TA3 < '7C' is not compliant with this specification.

*Basic response:* If T=1 is to be used, the ATR shall contain TA3 having a value in the range '7C' to 'FE' indicating an initial IFSC in the range 124 to 254 bytes.

*Terminal behaviour:* The terminal shall accept an ATR not containing TA3 (provided that b5 of TD2 is set to 0), and shall continue the card session using a value of '20' for TA3. The terminal shall reject an ATR containing TA3 having a value in the range '00' to '0F' or a value of 'FF'. The terminal shall accept an ATR containing TA3 having a value in the range '10' to 'FE'

The basic response coding of character TA3 is shown in Table 23:

**Table 23: Basic Response Coding of Character TA3**

| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|---|
| T=1 | x | x | x | x | x | x | x | x |
| | '00' to '7B' and 'FF' not allowed | | | | | | | |

#### 8.3.3.10 TB3

TB3 (if T=1 is indicated in TD2) indicates the values of the CWI and the BWI used to compute the CWT and BWT respectively. The l.s. nibble (b1–b4) is used to indicate the value of CWI, whilst the m.s. nibble (b5–b8) is used to indicate the value of BWI.

An ATR having CWI > '1' is not compliant with this specification.

**Note:** It is recommended that CWI = '0' for optimum performance.

*Basic response:* If T=1 is to be used, the ATR shall contain TB3 having the l.s. nibble '0' or '1', and the m.s. nibble in the range '0' to '4', indicating values of 0 or 1 for CWI and 0 to 4 for BWI.

The basic response coding of character TB3 is shown in Table 24:

**Table 24: Basic Response Coding of Character TB3**

| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|---|---|---|---|---|---|---|---|---|
| T=1 | 0 | x | x | x | 0 | 0 | 0 | y |
| | xxx is in the range 000 to 100<br>y is 0 or 1 | | | | | | | |

*Terminal behaviour:* The terminal shall reject an ATR not containing TB3, or containing a TB3 indicating BWI greater than 4 and/or CWI greater than 5, or having a value such that $2^{CWI} < (N + 1)$. It shall accept an ATR containing a TB3 having any other value.

Date for application:

| | | |
|---|---|---|
| Until January 2026 | Approved terminals due for renewal | may reject an ATR having a value such that $2^{CWI} = (N + 1)$ |
| From January 2026 | Approved terminals due for renewal | shall implement the requirements defined in the terminal behaviour above. |

**Note:** N is the extra guardtime indicated in TC1. When using T=1, if TC1='FF', the value of N shall be taken as –1. If TC1 is not contained in the ATR or TC1='00' the value of CWI shall be 1 in order to avoid the interpretation of a conflict between TC1 and TB3 by legacy terminals.

#### 8.3.3.11 TC3

TC3 (if T=1 is indicated in TD2) indicates the type of block error detection code to be used. The type of code to be used is indicated in b1, and b2 to b8 are not used.

*Basic response:* The ATR shall not contain TC3, thus indicating longitudinal redundancy check (LRC) as the error code to be used.

*Terminal behaviour:* The terminal shall accept an ATR containing TC3 = '00'. It shall reject an ATR containing TC3 having any other value.

### 8.3.4 TCK - Check Character

TCK has a value that allows the integrity of the data sent in the ATR to be checked. The value of TCK is such that the exclusive-ORing of all bytes from T0 to TCK inclusive is null.

An ATR shall contain TCK if a protocol type other than T=0 is indicated and shall not contain TCK if T=0 only is indicated.

From January 2023, ICCs returning an ATR different than explained above are not compliant to this specification.

*Basic responses:* The ATR shall not contain TCK if T=0 only is indicated. If a protocol type other than T=0 is indicated TCK shall be returned in the ATR.

*Terminal behaviour:*

The terminal shall be able to evaluate TCK when appropriately returned. It shall accept an ICC returning an ATR:

- containing a correct TCK if a protocol type other than T=0 is indicated, and
- not containing TCK if T=0 only is indicated.

The terminal shall reject an ICC returning an ATR:

- not containing TCK if a protocol type other than T=0 is indicated, or
- containing an incorrect TCK.

## 8.4 Terminal Behaviour during Answer to Reset

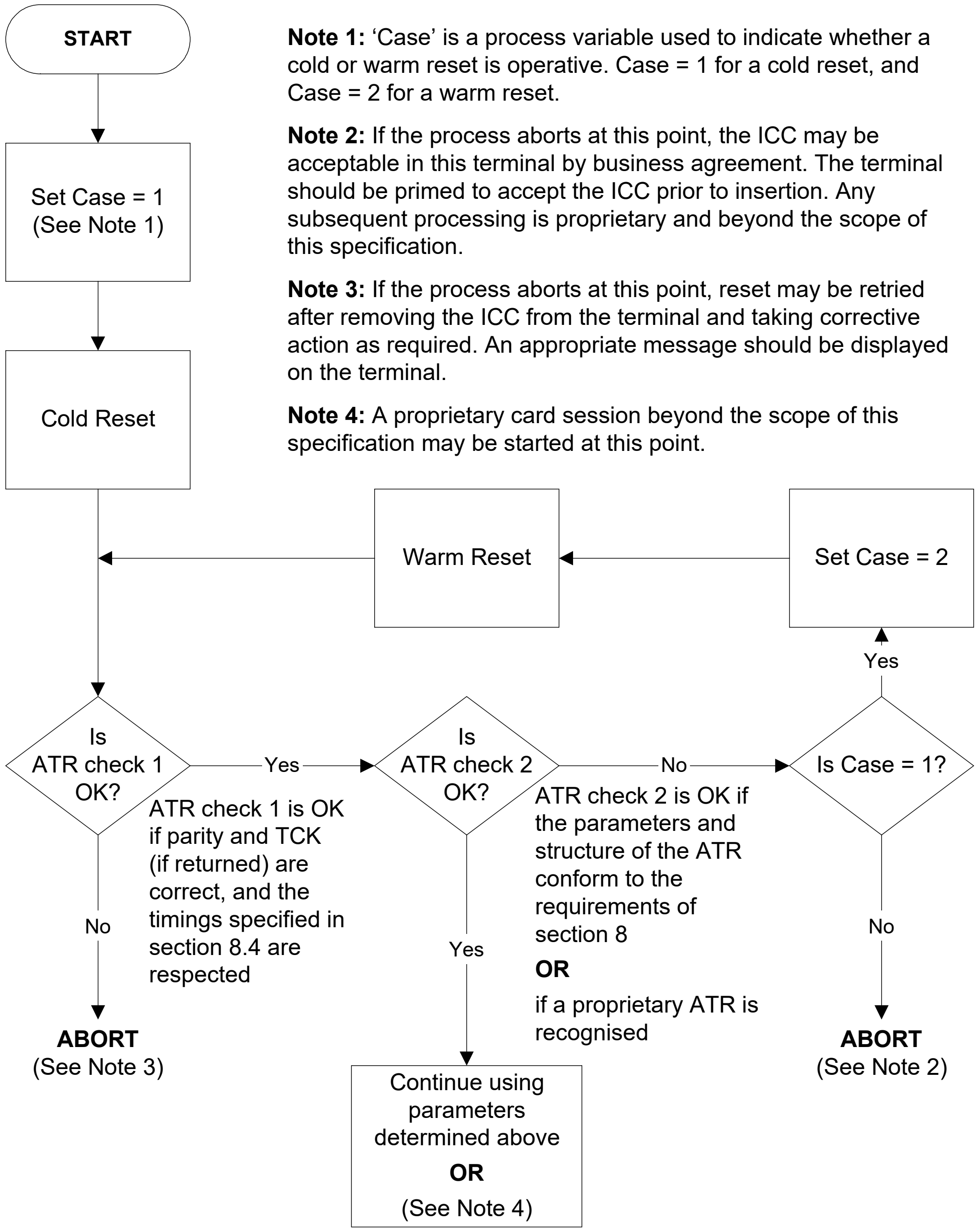
Following activation of the ICC contacts as described in section 6.1.2 the terminal shall initiate a cold reset as described in section 6.1.3.1. Subsequently the following shall apply:

- If the terminal rejects the ICC as described in section 8.3, it shall initiate the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the ATR.
- If the terminal rejects a cold ATR as described in section 8.3, it shall not immediately abort the card session but shall initiate a warm reset within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the cold ATR to the time that RST is set low. The terminal shall not initiate the warm reset until the T0 character has been received.
- If the terminal rejects a warm ATR as described in section 8.3, it shall initiate the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the warm ATR.
- The terminal shall be able to receive an ATR having a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 initial etus.
- The terminal shall be able to receive an ATR having maximum interval between two consecutive characters of 10,080 initial etus (9,600 + 480 initial etus). If a character is not received, the terminal shall abort the card session by initiating the deactivation sequence within 14,400 initial etus (9,600 + 4,800 initial etus) following the leading edge of the start bit of the last received character (the character from which timeout occurred).
- The terminal shall be able to receive an ATR having a duration of less than or equal to 20,160 initial etus. If the ATR (warm or cold) is not completed the terminal shall abort the card session by initiating the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) following the leading edge of the start bit of the TS character.
- If the terminal detects a parity error in a character returned in the ATR, it shall initiate the deactivation sequence within 24,000 initial etus (19,200 + 4,800 initial etus) measured from the leading edge of the start bit of the TS character of the ATR.
- Upon receipt of a valid cold or warm reset complying with the timings described above, the terminal shall proceed with the card session using the returned parameters. It may continue the card session as soon as the last character of the valid ATR (as indicated by the bit map characters T0 and/or TDi) and TCK, if present, has been received. Before transmitting, it shall wait at least the guardtime applicable to the protocol to be used (16 etus for T=0, BGT for T=1) measured from the leading edge of the start bit of the last character of the valid ATR.

## 8.5 Answer to Reset - Flow at the Terminal

Figure 10 illustrates an example of the process of an ICC returning an ATR to the terminal and the checks performed by the terminal to ensure conformance to section 8.

**Figure 10: ATR - Example Flow at the Terminal**




**Note 1:** 'Case' is a process variable used to indicate whether a cold or warm reset is operative. Case = 1 for a cold reset, and Case = 2 for a warm reset.

**Note 2:** If the process aborts at this point, the ICC may be acceptable in this terminal by business agreement. The terminal should be primed to accept the ICC prior to insertion. Any subsequent processing is proprietary and beyond the scope of this specification.

**Note 3:** If the process aborts at this point, reset may be retried after removing the ICC from the terminal and taking corrective action as required. An appropriate message should be displayed on the terminal.

**Note 4:** A proprietary card session beyond the scope of this specification may be started at this point.

# 9 Transmission Protocols

This section defines the structure and processing of commands initiated by the terminal for transmission control and for specific control in asynchronous half duplex transmission protocols.

Two types of protocol are defined, character protocol (T=0) and block protocol (T=1). ICCs shall support either protocol T=0 or protocol T=1. Terminals shall support both protocol T=0 and T=1. The protocol to be used for subsequent communication between the ICC and terminal is indicated in TD1, and shall be T=0 or T=1. If TD1 is absent in the ATR, T=0 is assumed. The protocol indicated by the ICC applies immediately after the answer to reset, as there is no PPS procedure. Other parameters returned in the ATR and relevant to a specific protocol are defined in sections 9.2 through 9.4.

The protocols are defined according to the following layering model:

- Physical layer, which describes the exchanges of bits and is common to both protocols.
- Data link layer, which includes the following sub-definitions:
    - Character frame, defining the exchanges of characters common to both protocols.
    - Character protocol T=0, defining the exchange of characters specific to T=0.
    - Error detection and correction specific to T=0.
    - Block protocol T=1, defining the exchanges of blocks specific to T=1.
    - Error detection and correction specific to T=1.
- Transport layer, which defines the transmission of application-oriented messages specific to each protocol.
- Application layer, which defines the exchange of messages according to an application protocol that is common to both transmission protocols.

## 9.1 Physical Layer

Both protocols T=0 and T=1 use the physical layer and character frame as defined in section 7.

## 9.2 Data Link Layer

This section describes the timing, specific options, and error handling for protocols T=0 and T=1.

### 9.2.1 Character Frame

The character frame as defined in section 7.2 applies to all messages exchanged between the ICC and the terminal.

### 9.2.2 Character Protocol T=0

#### 9.2.2.1 Specific Options - Character Timing for T=0

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the terminal to the ICC shall be between 12 and 266 etus as determined by the value of TC1 returned at the answer to reset (see sections 8.2 and 8.3). This interval may be less than the minimum interval of 16 etus allowed between two characters sent in opposite directions. If the value returned in TC1 is N (if TC1='FF' then N shall be taken as 0), the ICC shall be able to correctly interpret characters sent by the terminal with a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 + N etus.

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the ICC to the terminal shall be 12 etus. The terminal shall be able to correctly interpret characters sent by the ICC with a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 etus.

The maximum interval between the leading edges of the start bits of any two consecutive characters sent by the terminal shall not exceed 13 + N etus where N is the value returned in TC1 (if TC1='FF' then N shall be taken as 0).

The maximum interval between the leading edges of the start bits of any two consecutive characters sent by the ICC, other than a '60' procedure byte previously sent by the ICC and characters repeated due to the indication of a parity error, shall not exceed 13 etus.

The maximum interval between the leading edge of the start bit of any character sent by the ICC and the leading edge of the start bit of the previous character sent by the terminal or the previous '60' procedure byte sent by the ICC (the Work Waiting Time, or WWT) shall not exceed 960 x D x WI etus (D and WI are returned in TA1 and TC2, respectively).

The terminal shall be able to correctly interpret a character sent by the ICC with a maximum interval between the leading edge of the start bit of the character and the leading edge of the start bit of the previous character sent either by the ICC or the terminal of {WWT + (D x 480)} etus. If no character is received, the terminal shall initiate the deactivation sequence within {WWT + (D x 9600)} etus following the leading edge of the start bit of the character from which the timeout occurred.

For the ICC or terminal, the minimum interval between the leading edges of the start bits of the last character received and the first character sent in the opposite direction shall be 16 etus. The ICC or terminal shall be able to correctly interpret a character received within 15 etus timed from the leading edge of the start bit of the last character sent to the leading edge of the start bit of the received character. These timings do not apply during character repetition.

#### 9.2.2.2 Command Header

A command is always initiated by the terminal application layer (TAL) which sends an instruction via the TTL to the ICC in the form of a five byte header called the command header. The command header is comprised of five consecutive bytes, CLA, INS, P1, P2, and P3, where:

- CLA is the command class.
- INS is the instruction code.
- P1 and P2 contain additional instruction-specific parameters.
- P3 indicates either the length of data to be sent with the command to the ICC, or the maximum length of data expected in the response from the ICC, depending on the coding of INS.

These bytes together with any data to be sent with the command constitute the command transport protocol data unit (C-TPDU) for T=0. The mapping of the command application protocol data unit (C-APDU) onto the C-TPDU is described in section 9.3.

The TTL transmits the five-byte header to the ICC and waits for a procedure byte.

#### 9.2.2.3 Command Processing

Following reception of a command header by the ICC, the ICC shall return a procedure byte or status bytes SW1 SW2 (hereafter referred to as 'status') to the TTL. Both the TTL and ICC shall know implicitly at any point during exchange of commands and data between the TTL and the ICC what the direction of data flow is and whether it is the TTL or the ICC that is driving the I/O line.

#### 9.2.2.3.1 Procedure Byte

The procedure byte indicates to the TTL what action it shall take next. The coding of the byte and the action that shall be taken by the TTL is shown in Table 25.

**Table 25: Terminal Response to Procedure Byte**

| Procedure Byte Value | Action |
|---|---|
| Equal to INS byte | All remaining data bytes shall be transferred by the TTL, or the TTL shall be ready to receive all remaining data bytes from the ICC |
| Equal to complement of INS byte ($\overline{\mathrm{INS}}$) | The next data byte shall be transferred by the TTL, or the TTL shall be ready to receive the next data byte from the ICC |
| '60' | The TTL shall provide additional work waiting time as defined in this section |
| '61' | The TTL shall wait for a second procedure byte then send a GET RESPONSE command header to the ICC with a maximum length of 'xx', where 'xx' is the value of the second procedure byte |
| '6C' | The TTL shall wait for a second procedure byte then immediately resend the previous command header to the ICC using a length of 'xx', where 'xx' is the value of the second procedure byte |

In all cases, after the action has taken place the TTL shall wait for a further procedure byte or status.

#### 9.2.2.3.2 Status Bytes

The status bytes indicate to the TTL that command processing by the ICC is complete. The meaning of the status bytes is related to the command being processed. The coding of the first status byte and the action that shall be taken by the TTL are shown in Table 26.

**Table 26: Status Byte Coding**

| First Status Byte Value | Action |
|---|---|
| '6x' or '9x' (except '60', '61' and '6C') - status byte SW1 | TTL shall wait for a further status byte (status byte SW2) |

Following receipt of the second status byte, the TTL shall return the status bytes (together with any appropriate data - see section 9.3.1) to the TAL in the response APDU (R-APDU) and await a further C-APDU.

#### 9.2.2.4 Transportation of C-APDUs

A C-APDU containing only command data to be sent to the ICC, or only expecting data in response from the ICC (cases 2 and 3 in section 9.4), is mapped without change onto a T=0 C-TPDU. A C-APDU that contains and expects no data, or a C-APDU that requires data transmission to and from the ICC (cases 1 and 4 in section 9.4) is translated according to the rules defined in section 9.3 for transportation by a C-TPDU for T=0.

### 9.2.3 Error Detection and Correction for T=0

This procedure is mandatory for T=0 but does not apply during the answer to reset.

If a character is received with a parity error, the receiver shall indicate an error by setting the I/O line to state L at time (10.5 ± 0.2) etus following the leading edge of the start bit of the character for a minimum of 1 etu and a maximum of 2 etus.

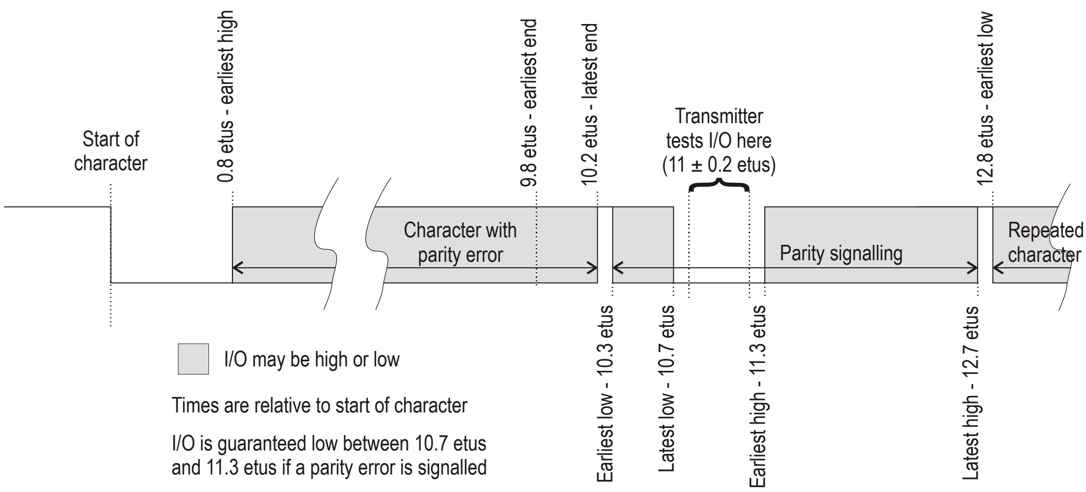
The transmitter shall test the I/O line (11 ± 0.2) etus after the leading edge of the start bit of a character was sent, and assumes that the character was correctly received if the I/O line is in state H.

If the transmitter detects an error, it shall repeat the disputed character after a delay of at least 2 etus following detection of the error. The transmitter shall repeat the same disputed character a maximum of three more times, and shall therefore send a character up to a maximum of five times in total (the original transmission followed by the first repeat and then three further repeats) in an attempt to achieve error free transmission.

If the last repetition is unsuccessful, the terminal shall initiate the deactivation sequence within (D x 960) etus following reception of the leading edge of the start bit of the invalid character (if it is the receiver), or within (D x 960) etus following detection of the signalling of the parity error by the ICC (if it is the transmitter).

Character repetition timing is illustrated in Figure 11.

**Figure 11: Character Repetition Timing**



When awaiting a procedure byte or status byte, if the byte returned by the ICC has a value other than specified in sections 9.2.2.3.1 and 9.2.2.3.2, the terminal shall initiate the deactivation sequence within 9,600 etus following the leading edge of the start bit of the (invalid) byte received.

## 9.2.4 Block Protocol T=1

The protocol consists of blocks transmitted between the TAL and the ICC to convey command and R-APDUs and transmission control information (for example, acknowledgment).

The data link layer block frame structure, specific options of the protocol, and protocol operations (including error handling) are defined below.

### 9.2.4.1 Block Frame Structure

The character frame as defined in section 7.2 applies.

The block is structured as illustrated in Table 27:

**Table 27: Structure of a Block**

| **Prologue Field - Mandatory -** | | | **Information Field - Optional -** | **Epilogue Field - Mandatory -** |
|---|---|---|---|---|
| **Node Address (NAD)** | **Protocol Control Byte (PCB)** | **Length (LEN)** | **APDU or Control Information (INF)** | **Error Detection Code (EDC)** |
| 1 byte | 1 byte | 1 byte | 0–254 bytes | 1 byte |

#### 9.2.4.1.1 Prologue Field

The Prologue Field is mandatory and consists of three mandatory bytes:

- Node address (NAD) to identify source and intended destination of the block
- Protocol control byte (PCB) to control data transmission
- Length (LEN) of the optional information field

***Node Address***

NAD is mandatory. Bits b1–b3 of NAD indicate the source node address (SAD) of the block, whilst bits b5–b7 indicate the intended destination node address (DAD) of the block. Bits b4 and b8 are unused and shall be set to 0.

These specifications do not support node addressing. The first block sent by the terminal following the ATR and all following blocks transmitted by either the terminal or ICC shall have the NAD = '00'.

If during the card session the terminal or ICC receives a block with a NAD ≠ '00', it may treat the block as invalid. In this event, it shall apply the error detection and correction techniques described in section 9.2.5.

***Protocol Control Byte***

The PCB is mandatory and indicates the type of block. There are three types of blocks, as defined in Table 28:

**Table 28: Types of Blocks**

| Type of Block | Short Name | Purpose |
|---|---|---|
| Information block | I-block | to convey APDUs |
| Receive-ready block | R-block | to convey acknowledgments (ACK or NAK) |
| Supervisory block | S-block | to exchange control information |

The coding of the PCB depends on its type and is defined in Table 29, Table 30, and Table 31.

**Table 29: Coding of the PCB of an I-block**

| | |
|---|---|
| b8 | 0 |
| b7 | Sequence number |
| b6 | Chaining (more data) |
| b5–b1 | Reserved for future use (RFU) |

**Table 30: Coding of the PCB of a R-block**

| | |
|---|---|
| b8 | 1 |
| b7 | 0 |
| b6 | 0 |
| b5 | Sequence number |
| b4–b1 | 0 = Error free<br>1 = EDC and/or parity error<br>2 = Other error(s)<br>Other values RFU |

**Table 31: Coding of the PCB of a S-block**

| | |
|---|---|
| b8 | 1 |
| b7 | 1 |
| b6 | 0 = Request<br>1 = Response |
| b5–b1 | 0 = Resynchronisation request<br>1 = Information field size request<br>2 = Abort request<br>3 = Extension of BWT request<br>Other values RFU |

***Length***

The Length (LEN) is mandatory and indicates the length of the INF part of the block; it may range from 0 to 254 depending on the type of block.

**Note:** This specification does not support I-blocks with LEN = 0.

#### 9.2.4.1.2 Information Field

The Information Field (INF) is conditional.

- When present in an I-block, it conveys application data.
- When present in a S-block, it conveys control information.
- A R-block shall not contain an INF.

#### 9.2.4.1.3 Epilogue Field

The Epilogue Field is mandatory and contains the EDC of the transmitted block. A block is invalid when a parity error and/or an EDC error occurs. This specification supports only the LRC as EDC. The LRC is one byte in length and is calculated as the exclusive-OR of all the bytes starting with the NAD and including the last byte of INF, if present.

**Note:** TCi ($i > 2$), which indicates the type of error detection code to be used, is not returned by the ICC in the ATR. The normal default of the LRC is thus used for the EDC.

#### 9.2.4.1.4 Block Numbering

I-blocks are numbered using a modulo-2 number coded on one bit. The numbering system is maintained independently at the ICC and the terminal as senders. The value of the number starts with zero for the first I-block sent after the answer to reset by a sender and is incremented by one after sending each I-block. The number is reset to zero by the sender after resynchronisation.

R-blocks are numbered using a modulo-2 number coded on one bit. A R-block is used to acknowledge a chained I-block or to request retransmission of an invalid block. In either case, b5 of the PCB of the R-block carries the sequence number of the next I-block its sender expects to receive.

A S-block carries no number.

### 9.2.4.2 Specific Options

This section defines the information field sizes and timings to be used with protocol type T=1.

#### 9.2.4.2.1 Information Field Sizes

The IFSC is the maximum length of the information field of blocks that can be received by the ICC and is defined as follows. At the answer to reset the IFSI is returned by the ICC in TA3 indicating the size of the IFSC that can be accommodated by the ICC. IFSI may take values in the range '10' to 'FE' that indicate values for IFSC in the range 16 to 254 bytes. The maximum block size that can be received by the ICC is therefore (IFSC + 3 + 1) bytes including the prologue and epilogue fields. The size established during the answer to reset shall be used throughout the rest of the card session or until a new value is negotiated by the ICC by sending a S(IFS request) block to the terminal.

The information field size for the terminal (IFSD) is the maximum length of the information field of blocks that can be received by the terminal. The initial size immediately following the answer to reset shall be 254 bytes, and this size shall be used throughout the rest of the card session.

#### 9.2.4.2.2 Timing for T=1

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the terminal to the ICC shall be between 11 and 42 etus as indicated by the value of TC1 returned at the answer to reset (see sections 8.2 and 8.3). If the value returned in TC1 is N (if TC1='FF' then N shall be taken as -1), the ICC shall be able to correctly interpret characters sent by the terminal with a minimum interval between the leading edges of the start bits of two consecutive characters of 11.8 + N etus.

The minimum interval between the leading edges of the start bits of two consecutive characters sent by the ICC to the terminal shall be 11 etus. The terminal shall be able to correctly interpret characters sent by the ICC with a minimum interval between the leading edges of the start bits of two consecutive characters of 10.8 etus.

The maximum interval between the leading edges of the start bits of two consecutive characters sent in the same block (the character waiting time, CWT) shall not exceed ($2^{CWI}$ + 11) etus. The character waiting time integer, CWI shall have a value of 0 or 1 as described in section 8.3.3.10, and thus CWT is 12 or 13 etus. Terminals shall be able to correctly interpret CWI up to 5. The receiver shall be able to correctly interpret a character having a maximum interval between the leading edge of the start bit of the character and the leading edge of the start bit of the previous character of (CWT + 4) etus.

The maximum interval between the leading edge of the start bit of the last character that gave the right to send to the ICC and the leading edge of the start bit of the first character sent by the ICC (the block waiting time, BWT) shall not exceed $\{(2^{BWI} \times 960) + 11\}$ etus. The block waiting time integer, BWI shall have a value in the range 0 to 4 as described in section 8.3.3.10, and thus BWT lies in the range 971 to 15,371 etus for a D of 1.

The terminal shall be able to correctly interpret the first character of a block sent by the ICC following a time BWT + (D x 960) etus.

For the ICC or terminal, the minimum interval between the leading edges of the start bits of the last received character and the first character sent in the opposite direction (the block guard time, BGT) shall be 22 etus. The ICC or terminal shall be able to correctly interpret a character received within 21 etus timed from the leading edge of the start bit of the last character that it sent to the leading edge of the start bit of the received character.

**Note:** In general, for values of FI and DI other than 1, BWT is calculated using the formula:

$$\mathrm{BWT} = \left(\left(2^{\mathrm{BWI}} \times 960 \times \frac{372\mathrm{D}}{\mathrm{F}}\right) + 11\right) \mathrm{etu}$$

### 9.2.4.3 Error Free Operation

The protocol rules for error free operation are as follows:

1. The first block transmitted after the answer to reset shall be sent by the terminal to the ICC and shall be a S(IFS request) block with PCB = 'C1' and with IFSD = 254 (value indicated in the single byte INF field). No further S(IFS request) blocks shall be sent by the terminal during the card session.
2. The ICC shall return a S(IFS response) block to the terminal acknowledging the change to the size of the IFSD. The PCB of the S(IFS response) block sent in response shall have the value 'E1', and the INF field shall have the same value as the INF field of the block requesting the change.
3. The ICC shall not send an S(IFS request) block to change the size of the IFSC from the initial value indicated at the answer to reset. The terminal shall support a S(IFS request) block sent by a card that wishes to change the size of the IFSC. The PCB of the S(IFS request) block has the value 'C1' indicating a request to change the IFSC. The INF field shall contain a byte the value of which indicates the size in bytes of the requested new IFSC. The terminal shall support an INF byte in the range '10' to 'FE' and return a S(IFS response) block to the ICC acknowledging the change to the size of the IFSC. The PCB of the S(IFS response) block sent in response shall have the value 'E1', and the INF field shall have the same value as the INF field of the block requesting the change.
4. During the card session, only blocks as defined in this section shall be exchanged. The half duplex block protocol consists of blocks transmitted alternately by the terminal and the ICC. When the sender has transmitted a complete block, the sender switches to the receiving state.

5. When the receiver has received the number of characters in accordance with the value of LEN and the EDC, the receiver gains the right to send.
6. The ICC shall acknowledge an I-block transmitted by the terminal. The acknowledgment is indicated in the sequence number of the I-block, or the R-block if chaining is in use (except the last block of the chain), that the ICC returns to the terminal.
7. A non-chained I-block or the last I-block of a chain is considered by the sender to be acknowledged when the sequence number of the I-block received in response differs from the sequence number of the previously received I-block. If no I-block was previously received, the sequence number of the I-block sent in response shall be 0.
8. When a R-block is received, b5 shall be evaluated. The receiver is not required to evaluate bits b4–b1 of the PCB. Optional evaluation of bits b4–b1 shall not result in any action which contradicts the protocol rules defined in this specification
9. During chaining, a chained I-block (except the last I-block of a chain) is considered by the sender to be acknowledged when the sequence number of the R-block sent in response differs from the sequence number of the I-block being acknowledged.
10. If the ICC requires more than the BWT to process the previously received I-block, it shall send a waiting time extension request S(WTX request) block, where the INF contains the one-byte binary integer multiplier of the BWT value requested. The terminal shall acknowledge by sending a waiting time extension response S(WTX response) block with the same value in the INF. The time allocated (which is the time requested in the S(WTX request) block, and which replaces BWT for this instance only) starts at the leading edge of the last character of the S(WTX response) block. After the ICC responds, BWT is again used as the time allowed for the ICC to process the I-block.
11. S-blocks are used only in pairs. A S(request) block is always followed by a S(response) block.

When synchronisation as outlined above is lost, the procedure described in section 9.2.5 shall apply.

#### 9.2.4.4 Chaining

When the sender has to transmit data of length greater than IFSC or IFSD bytes, it shall divide it into several consecutive I-blocks. The transmission of these multiple I-blocks is achieved using the chaining function described below.

The chaining of I-blocks is controlled by b6 of the PCB. The coding of b6 is as follows:

- b6 = 0 Last block of the chain
- b6 = 1 Subsequent block follows

Any I-block with b6 = 1 shall be acknowledged by a R-block according to section 9.2.4.1.

The last block of a chain sent by the terminal shall be acknowledged by either an I-block if correctly received, or a R-block if incorrectly received. The last block of a chain sent by the ICC shall be acknowledged by a R-block if incorrectly received; if correctly received, the terminal will only transmit further I-blocks if another command is to be processed.

##### 9.2.4.4.1 Rules for Chaining

The TTL shall support chaining for both transmitted and received blocks. The ICC may optionally chain blocks sent to the terminal. Chaining is only possible in one direction at a time. The following rules for chaining apply:

- When the terminal is the receiver, the terminal shall accept a sequence of chained I-blocks sent from the ICC of length ≤ IFSD bytes per block.
- When the ICC is the receiver, the ICC shall accept a sequence of chained I-blocks sent from the terminal all having length LEN = IFSC except the last block, whose length may be in the range 1 to IFSC bytes inclusive.
- When the ICC is the receiver, if an I-block sent by the terminal has length > IFSC, the ICC shall reject it using a R-block.
- If the ICC as sender chains blocks sent to the terminal it shall send I-blocks of length ≤IFSD bytes per block.
- When the terminal is the sender, all I-blocks of a chain sent to the ICC shall have LEN = IFSC bytes except the last, which shall have a length in the range 1 to IFSC bytes inclusive.
- During chaining, the ICC shall not attempt to negotiate a new IFSC by sending a S(IFSC request) block to the terminal.

#### 9.2.4.4.2 Construction of Chained Blocks

C-APDUs are transported from the TTL to the ICC in the INF field of I-blocks (see section 9.3.2). If a C-APDU is too large to fit in one block, it is chained over several as illustrated in Figure 12.

**Figure 12: Chaining C-APDU**

| | | | | |
|---|---|---|---|---|
| Block(1) | CLA INS P1 P2 | Lc | Data | Data |
| Block(2) | Data | | Data | Data |
| Block(n) | Data | | Data | Le |

The data and status returned by the ICC may optionally be chained over several I-blocks as illustrated in Figure 13.

**Figure 13: Chaining I-Blocks**

| | | | |
|---|---|---|---|
| Block(1) | Data | Data | Data |
| Block(2) | Data | Data | Data |
| Block(n) | Data | Data | SW1 SW2 |

**Note:** The above examples are for a case 4 command and show only the INF fields of the chained blocks. Each block also has a prologue and epilogue field. All chained blocks shall contain an INF field having a length in the range 1 to IFSD bytes if the ICC is the sender, or IFSC bytes during chaining and 1 to IFSC bytes in the last block of the chain if the terminal is the sender.

### 9.2.5 Error Detection and Correction for T=1

The following errors shall be detected by the TTL:

- Transmission error including parity error, EDC error, and BWT time-out.
- Loss of synchronisation assumed when the actual block size is inconsistent with the size indicated by the value in LEN.
- Protocol error (infringement of the rules of the protocol).
- Abort request for a chain of blocks.

If a parity error is detected, character repetition shall not be implemented when using T=1.

Error recovery is attempted in the following manner.

The TTL shall attempt error recovery by trying the following techniques in the order shown.

1. Retransmission of blocks

Deactivation of the ICC contacts

The ICC shall attempt error recovery by trying retransmission of blocks.

If a block is retransmitted, the retransmitted block shall be identical to the originally transmitted block.

**Note:** In some terminals the TTL may not be solely responsible for error handling. Where 'TTL' is used it includes any functionality present in the terminal as applicable.

The following types of block are considered invalid:

- Blocks containing transmission errors, i.e. parity/EDC incorrect
- Blocks that have formatting errors, i.e. blocks constructed incorrectly by the sender (syntax error)
- Blocks that are unexpected according to the rules of the protocol at any particular point in an exchange, for example, a S(Response) block received in response to an I-block.

A R-block received indicating an error condition is not an invalid block.

### 9.2.6 Protocol Rules for T=1 Error Handling

The following rules apply for error handling and correction. In each case where a R-block is sent, the error coding bits b4–b1 may optionally be evaluated, but shall not result in any action which contradicts the protocol rules defined in this specification.

1. If the first block received by the ICC after the answer to reset is invalid, it shall return a R-block to the TTL with b5 = 0 and NAD = 0.
2. If there is no response from the ICC to a block sent by the TTL, the terminal shall:

   (a) initiate the deactivation sequence

   OR

   (b) if the block not responded to was an I-block, R-block, or S(Response) block, transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4

   OR

   (c) if the block not responded to was a S(Request) block, retransmit the S(Request) block

   between {BWT + (D x 960)} and {BWT + (D x 4,800)} etus (or between {WTX + (n x D x 960)} and {WTX + (n x D x 4,800)} etus if a waiting time extension has been negotiated) from the leading edge of the start bit of the last character of the block to which there was no response.
3. If during reception of a block by the terminal an expected character is not received, the terminal shall:

   (a) initiate the deactivation sequence

   OR

   (b) if the block not responded to was an I-block, R-block, or S(Response) block, transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4

   OR

   (c) if the block not responded to was a S(Request) block, retransmit the S(Request) block

   within (CWT + 4) and (CWT + 4,800) etus from the leading edge of the start bit of the last character received.
4. If an invalid block is received in response to an I-block, the sender shall transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4.
5. If an invalid block is received in response to a R-block, the sender shall retransmit the R-block.
6. If a correct S(... response) block is not received in response to a S(... request) block, the sender shall retransmit the S(... request) block.

7. If an invalid block is received in response to a S(... response) block, the sender shall transmit a R-block with its sequence number coded as specified in section 9.2.4.1.4.
8. If the TTL has sent three consecutive blocks of any type without obtaining a valid response, it shall initiate the deactivation sequence within {BWT + (D x 14,400)} etus following the leading edge of the start bit of the last character of the block requesting retransmission.

   **Note:** Resynchronisation is not required by this specification. If for proprietary reasons the terminal supports resynchronisation, it may attempt this by sending a S(RESYNCH request) block, then behave as specified in ISO/IEC 7816-3.

   If the ICC has sent a block a maximum of twice in succession (the original transmission followed by one repeat) without obtaining a valid response, it shall remain in reception mode.
9. A S(ABORT request) shall not be sent by the terminal. If the terminal receives a S(ABORT request) from the ICC, it shall terminate the card session by initiating the deactivation sequence within (D x 9,600) etus following reception of the leading edge of the start bit of the last character of the S(ABORT request) block.

   **Note:** Transaction abortion is not required by this specification. If an ICC or terminal supports abortion for proprietary reasons, it may issue a S(ABORT request), but note that it will receive an invalid response if the receiver does not support abortion. In this event, the card session will be terminated according to the rules above. If a terminal optionally supporting abortion receives a S(ABORT request) from an ICC, it may return a S(ABORT response) rather than terminating the card session.

## 9.3 Terminal Transport Layer (TTL)

This section describes the mechanism by which command and response APDUs are transported between the terminal and the ICC. APDUs are command or response messages, and since both command and response messages may contain data, the TTL shall be capable of managing the four cases defined in section 9.4. The construction of C-APDUs and R-APDUs are described in sections 9.4.1 and 9.4.2, respectively.

The C-APDU is passed from the TAL to the TTL where it is mapped in a manner appropriate to the transmission protocol to be used before being sent to the ICC. Following processing of the command by the ICC, data (if present) and status are returned by the ICC to the TTL, which maps it onto the R-APDU.

### 9.3.1 Transport of APDUs by T=0

This section describes the mapping of C-APDUs and R-APDUs, the mechanism for exchange of data between the TTL and the ICC, and the use of the GET RESPONSE command for retrieval of data from the ICC when case 2 or 4 commands are used.

#### 9.3.1.1 Mapping of C-APDUs and R-APDUs and Data Exchange

The mapping of the C-APDU onto the T=0 command header is dependent upon the case of the command. The mapping of the data (if present) and status returned by the ICC onto the R-APDU is dependent upon the length of the data returned and the meaning of the status bytes.

Procedure bytes '61xx' and '6Cxx' are returned by the ICC to control exchanges between the TTL and the ICC, and should never be returned to the TAL. Command processing in the ICC is not complete if it has returned procedure bytes '61xx' or '6Cxx'.

**Note:** For proprietary reasons, the TTL may in addition be capable of accepting data from the ICC without using the '61' and '6C' procedure bytes. Such functionality is not required and is beyond the scope of these specifications.

Normal status on completion of processing a command is indicated if the ICC returns status bytes SW1 SW2 = '9000' to the TTL. The TTL shall discontinue processing of a command (i.e. pass the R-APDU to the TAL and wait for a further C-APDU from the TAL) on receipt of any other status (but not on receipt of procedure bytes '61xx' and '6Cxx') from the ICC. (For case 4 commands only, immediately following successful transmission of command data to the ICC, the TTL shall continue processing the command if warning status bytes ('62xx' or '63xx') or application related status bytes ('9xxx' except '9000') are received.)

The following descriptions of the mapping of data and status returned by the ICC onto the R-APDU are for information, and apply only after the ICC has completed processing of the command, successfully or otherwise, and all data (if present) has been returned by the ICC under the control of '61xx' and '6Cxx' procedure bytes. Detailed use of the INS, $\overline{\text{INS}}$, and '60' procedure bytes is not described.

The status returned by the ICC shall relate to the most recently received command; where a GET RESPONSE command is used to complete the processing of a case 2 or case 4 command, any status returned by the ICC after receipt of the GET RESPONSE command shall relate to the GET RESPONSE command, not to the case 2 or case 4 command which it completes.

##### 9.3.1.1.1 Case 1

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and P3 of the T=0 command header is set to '00'.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header the ICC, under normal or abnormal processing, shall return status to the TTL.

   (The ICC shall analyse the T=0 command header to determine whether it is processing a case 1 command or a case 2 command requesting all data up to the maximum length available.)
3. On receipt of status from the ICC, the TTL shall discontinue processing of the command.

See Annex A1 for details of the exchanges between the TTL and the ICC.

The status returned to the TTL from the ICC after completion of processing of the command is mapped onto the mandatory trailer of the R-APDU without change.

#### 9.3.1.1.2 Case 2

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and length byte 'Le' from the conditional body of the C-APDU is mapped onto P3 of the T=0 command header. READ RECORD commands issued during application selection and all case 2 commands issued according to *EMV Integrated Circuit Card Specifications for Payment Systems – Book 3: Application Specification* shall have Le = '00'.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header:
   (a) under normal processing, the ICC shall return data and status to the TTL, using procedure bytes '6Cxx' (and if required, procedure bytes '61xx') to control the return of data

   OR

   (b) under abnormal processing, the ICC shall return status only to the TTL.
3. On receipt of the data (if present) and status from the ICC, the TTL shall discontinue processing the command.

See Annex A2 and Annex A5, for details of the exchanges between the TTL and the ICC, including use of the '61xx' and '6Cxx' procedure bytes.

The data (if present) and status returned to the TTL from the ICC after completion of processing of the command, or the status returned by the ICC that caused the TTL to discontinue processing of the command, are mapped onto the R-APDU as follows:

- The data returned (if present) is mapped onto the conditional body of the R-APDU. If no data is returned, the conditional body of the R-APDU is left empty.
- The status returned is mapped onto the mandatory trailer of the R-APDU without change.

##### 9.3.1.1.3 Case 3

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and length byte 'Lc' from the conditional body of the C-APDU is mapped onto P3 of the T=0 command header.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header:
   (a) If the ICC returns a procedure byte, the TTL shall send the data portion of the conditional body of the C-APDU to the ICC under the control of procedure bytes returned by the ICC.

   OR

   (b) If the ICC returns status, the TTL shall discontinue processing of the command.
3. If processing was not discontinued in step 2(b), the ICC shall return status following receipt of the conditional body of the C-APDU and completion of processing the command.
4. On receipt of status from the ICC, the TTL shall discontinue processing the command.

See Annex A3, for details of the exchanges between the TTL and the ICC.

The status returned to the TTL from the ICC after completion of processing of the command, or the status returned by the ICC that caused the TTL to discontinue processing of the command, is mapped onto the R-APDU without change.

#### 9.3.1.1.4 Case 4

The C-APDU header is mapped onto the first four bytes of the T=0 command header, and length byte 'Lc' from the conditional body of the C-APDU is mapped onto P3 of the T=0 command header. SELECT commands issued during application selection and all case 4 commands issued according to *EMV Integrated Circuit Card Specifications for Payment Systems – Book 3: Application Specification* shall have Le = '00'.

The flow of the exchange is as follows:

1. The TTL shall send the T=0 command header to the ICC.
2. On receipt of the command header:
   (a) If the ICC returns a procedure byte, the TTL shall send the data portion of the conditional body of the C-APDU to the ICC under the control of procedure bytes returned by the ICC.

   OR

   (b) If the ICC returns status, the TTL shall discontinue processing of the command.
3. If processing was not discontinued in step 2(b), following receipt of the conditional body of the C-APDU:
   (a) under normal processing, the ICC shall return procedure bytes '61xx' to the TTL requesting the TTL to issue a GET RESPONSE command to retrieve the data from the ICC.

   OR

   (b) under abnormal processing, the ICC shall return status only to the TTL.
4. On receipt of the procedure bytes or status returned in step 3:
   (a) If the ICC returned '61xx' procedure bytes as in step 3(a), the TTL shall send a GET RESPONSE command header to the ICC with P3 set to a value less than or equal to the value contained in the 'xx' byte of '61xx' procedure bytes.

   OR

   (b) If the ICC returned status as in step 3(b) that indicates a warning ('62xx' or '63xx'), or which is application related ('9xxx' but not '9000'), the TTL shall send a GET RESPONSE command with Le='00'.

   OR

   (c) If the ICC returned status as in step 3(b) other than that described in step 4(b), the TTL shall discontinue processing of the command.
5. If processing was not discontinued in step 4(c), the GET RESPONSE command shall be processed according to the rules for case 2 commands in section 9.3.1.1.2.

See Annex A4 and Annex A6, for details of the exchanges between the TTL and the ICC, including use of the '61xx' and '6Cxx' procedure bytes.

The data (if present) and status returned to the TTL from the ICC after completion of processing of the command, or the status returned by the ICC that caused the TTL to discontinue processing of the command, are mapped onto the R-APDU as follows:

- The data returned (if present) is mapped onto the conditional body of the R-APDU. If no data is returned, the conditional body of the R-APDU is left empty.
- The first status returned during processing of the entire case 4 command, including the GET RESPONSE command if used, is mapped onto the mandatory trailer of the R-APDU without change.

#### 9.3.1.2 Use of Procedure Bytes '61xx' and '6Cxx'

The ICC returns procedure bytes '61xx' and '6Cxx' to the TTL to indicate to it the manner in which it should retrieve the data requested by the command currently being processed. These procedure bytes are only used when processing case 2 and 4 commands.

Procedure bytes '61xx' instruct the TTL to issue a GET RESPONSE command to the ICC. P3 of the GET RESPONSE command header is set to ≤ 'xx'.

Procedure bytes '6Cxx' instruct the TTL to immediately resend the previous command header setting P3 = 'xx'.

Usage of these procedure bytes during error free processing with case 2 and 4 commands is as follows. In the case of an error, the ICC may return status indicating error or warning conditions instead of the '61xx' or '6Cxx' response.

#### 9.3.1.2.1 Case 2 Commands

1. If the ICC receives a case 2 command header and Le = '00' or Le > Licc, it shall return:
   (a) procedure bytes '6C Licc' instructing the TTL to immediately resend the command header with P3 = Licc

   OR

   (b) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

   **Note:** If Le = '00' and the ICC has 256 bytes of data to return, it should proceed as defined in the following rule for Le = Licc.

2. If the ICC receives a case 2 command header and Le = Licc, it shall return:
   (a) data of length Le (= Licc) under the control of the INS, $\overline{\text{INS}}$, or '60' procedure bytes followed by the associated status

   OR

   (b) procedure bytes '61xx' instructing the TTL to issue a GET RESPONSE command with a maximum length of 'xx'

   OR

   (c) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

3. If the ICC receives a case 2 command header and Le < Licc, it shall return:
   (a) data of length Le under the control of the INS, $\overline{\text{INS}}$, or '60' procedure bytes followed by procedure bytes '61xx' instructing the TTL to issue a GET RESPONSE command with a maximum length of 'xx'

   OR

   (b) procedure bytes '6C Licc' instructing the TTL to immediately resend the command header with P3 = Licc

   OR

   (c) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

3(b) above is not a valid response by the ICC to a GET RESPONSE command.

#### 9.3.1.2.2 Case 4 Commands

1. If the ICC receives a case 4 command, after processing the data sent with the C-APDU, it shall return:
   (a) procedure bytes '61 xx' instructing the TTL to issue a GET RESPONSE command with a maximum length of 'xx'

   OR

   (b) status indicating a warning or error condition (but not SW1 SW2 = '90 00').

The GET RESPONSE command so issued is then treated as described in section 9.3.1.2.1 for case 2 commands.

#### 9.3.1.3 GET RESPONSE Command

The GET RESPONSE command is issued by the TTL to obtain available data from the ICC when processing case 2 and 4 commands. It is employed only when the T=0 protocol type is in use.

The structure of the command message is shown in Table 32:

**Table 32: Structure of Command Message**

| | |
|---|---|
| CLA | '00' |
| INS | 'C0' |
| P1 | '00' |
| P2 | '00' |
| Le | Maximum length of data expected |

Following normal processing, the ICC returns status bytes SW1 SW2 = '9000' and Licc bytes of data.

In the event that an error condition occurs, the coding of the error status bytes (SW1 SW2) is shown in Table 33:

**Table 33: GET RESPONSE Error Conditions**

| SW1 | SW2 | Meaning |
|---|---|---|
| '62' | '81' | Part of returned data may be corrupted |
| '67' | '00' | Length field incorrect |
| '6A' | '86' | P1 P2 ≠ '00' |
| '6F' | '00' | No precise diagnosis |

### 9.3.2 Transportation of APDUs by T=1

The C-APDU is sent from the TAL to the TTL. The TTL maps the C-APDU onto the INF field of an I-block without change and sends the I-block to the ICC.

Response data (if present) and status are returned from the ICC to the TTL in the INF field of an I-block. If the ICC returns status indicating normal processing ('9000'), a warning ('62xx' or '63xx'), or which is application related ('9xxx'), it shall also return data (if available) associated with processing of the command.

No data shall be returned with any other status. The contents of the INF field of the I-block are mapped onto the R-APDU without change and returned to the TAL.

**Note:** C-APDUs and response data/status may be chained over the INF fields of multiple blocks if required.

## 9.4 Application Layer

The application protocol consists of an ordered set of exchanges between the TAL and the TTL. Each step in an application layer exchange consists of a command-response pair, where the TAL sends a command to the ICC via the TTL, and the ICC processes it and sends a response via the TTL to the TAL. Each specific command has a specific response. An APDU is defined as a command message or a response message.

Both command and response messages may contain data. Thus, four cases shall be managed by the transmission protocols via the TTL, as shown in Table 34:

**Table 34: Definition of Cases for Data in APDUs**

| Case | Command Data | Response Data |
|---|---|---|
| 1 | Absent | Absent |
| 2 | Absent | Present |
| 3 | Present | Absent |
| 4 | Present | Present |

### 9.4.1 C-APDU

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.

These mandatory header bytes are defined as follows:

- CLA: Instruction class; may take any value except 'FF'.
- INS: Instruction code within the instruction class. The INS is only valid if the l.s. bit is 0, and the m.s. nibble is neither '6' nor '9'.
- P1, P2: Reference bytes completing the INS.

The conditional body consists of a string of bytes defined as follows:

- 1 byte, denoted by Lc, defining the number of data bytes to be sent in the C-APDU. The value of Lc may range from 1 to 255.
- String of bytes sent as the data field of the C-APDU, the number of bytes sent being as defined by Lc.
- 1 byte, denoted by Le, indicating the maximum number of data bytes expected in the R-APDU. The value of Le may range from 0 to 255; if Le = 0, the maximum number of bytes expected in the response is 256.

The four possible C-APDU structures are defined in Table 35:

**Table 35: C-APDU Structures**

| Case | Structure |
|---|---|
| 1 | CLA INS P1 P2 |
| 2 | CLA INS P1 P2 Le |
| 3 | CLA INS P1 P2 Lc Data |
| 4 | CLA INS P1 P2 Lc Data Le |

### 9.4.2 R-APDU

The R-APDU is a string of bytes consisting of a conditional body followed by a mandatory trailer of two bytes denoted SW1 SW2.

The conditional body is a string of data bytes with a maximum length as defined by Le in the C-APDU.

The mandatory trailer indicates the status of the ICC after processing the command.

# Annex A Examples of Exchanges Using T=0

The following examples illustrate exchanges of data and procedure bytes between the TTL and ICC.

Note the following:

- The use of procedure bytes '60' and $\overline{\mathrm{INS}}$ is not illustrated.
- [Data(x)] means x bytes of data.
- Case 2 and 4 commands have Le = '00' requesting the return of all data from the ICC up to the maximum available. Le = '00' is used in these examples to illustrate typical exchanges that may be observed when executing the application specified in *EMV Integrated Circuit Card Specifications for Payment Systems – Book 3: Application Specification*, Le may take other values when executing a proprietary application.

Sections A1 to A4 illustrate typical exchanges using case 1 to 4 commands. Sections A5 and A6 illustrate the more extensive use of procedure bytes '61 xx' when used with case 2 and 4 commands. Section A7 illustrates a warning condition with a case 4 command.

## A1 Case 1 Command

A C-APDU of {CLA INS P1 P2} is passed from the TAL to the TTL (note that P3 of the C-TPDU is set to '00').

| TTL | ICC |
|---|---|
| [CLA INS P1 P2 00] ⇒ | |
| | ⇐ 90 00 |

A R-APDU of {90 00} is returned from the TTL to the TAL.

# A2 Case 2 Command

A C-APDU of {CLA INS P1 P2 00} is passed from the TAL to the TTL.

| TTL | ICC |
|---|---|
| [CLA INS P1 P2 00] ⇒ | |
| | ⇐ 6C Licc |
| [CLA INS P1 P2 Licc] ⇒ | |
| | ⇐ INS [Data(Licc)] 90 00 |

A R-APDU of {[Data(Licc)] 90 00} is returned from the TTL to the TAL.

# A3 Case 3 Command

A C-APDU of {CLA INS P1 P2 Lc [Data(Lc)]} is passed from the TAL to the TTL.

| TTL | ICC |
|---|---|
| [CLA INS P1 P2 Lc] ⇒ | |
| | ⇐ INS |
| [Data(Lc)] ⇒ | |
| | ⇐ 90 00 |

A R-APDU of {90 00} is returned from the TTL to the TAL.

# A4 Case 4 Command

A C-APDU of {CLA INS P1 P2 Lc [Data (Lc)] 00} is passed from the TAL to the TTL.

| TTL | ICC |
|---|---|
| [CLA INS P1 P2 Lc] ⇒ | |
| | ⇐ [INS] |
| [Data(Lc)] ⇒ | |
| | ⇐ 61 Licc |
| [00 C0 00 00 Licc] ⇒ | |
| | ⇐ C0 [Data(Licc)] 90 00 |

A R-APDU of {[Data(Licc)] 90 00} is returned from the TTL to the TAL.

# A5 Case 2 Command Using the '61' and '6C' Procedure Bytes

A C-APDU of {CLA INS P1 P2 00} is passed from the TAL to the TTL.

| TTL | ICC |
|---:|---|
| [CLA INS P1 P2 00] ⇒ | |
| | ⇐ 6C Licc |
| [CLA INS P1 P2 Licc] ⇒ | |
| | ⇐ 61 xx |
| [00 C0 00 00 yy] ⇒ | |
| | ⇐ C0 [Data(yy)] 61 zz |
| [00 C0 00 00 zz] ⇒ | |
| | ⇐ C0 [Data(zz)] 90 00 |

Where yy ≤ xx

A R-APDU of {[Data(yy + zz)] 90 00} is returned from the TTL to the TAL.

# A6 Case 4 Command Using the '61' Procedure Byte

A C-APDU of {CLA INS P1 P2 Lc [Data Lc] 00} is passed from the TAL to the TTL.

| TTL | ICC |
|---:|---|
| [CLA INS P1 P2 Lc] ⇒ | |
| | ⇐ [INS] |
| [Data(Lc)] ⇒ | |
| | ⇐ 61 xx |
| [00 C0 00 00 xx] ⇒ | |
| | ⇐ C0 [Data(xx)] 61 yy |
| [00 C0 00 00 yy] ⇒ | |
| | ⇐ C0 [Data(yy)] 90 00 |

A R-APDU of {[Data(xx + yy)] 90 00} is returned from the TTL to the TAL.

# A7 Case 4 Command with Warning Condition

A C-APDU of {CLA INS P1 P2 Lc [Data Lc] 00} is passed from the TAL to the TTL.

| TTL | ICC |
|---|---|
| [CLA INS P1 P2 Lc] ⇒ | |
| | ⇐ [INS] |
| [Data(Lc)] ⇒ | |
| | ⇐ 62 xx |
| [00 C0 00 00 00] ⇒ | |
| | ⇐ 6C Licc |
| [00 C0 00 00 Licc] ⇒ | |
| | ⇐ C0 [Data(Licc)] 90 00 |

OR

| TTL | ICC |
|---|---|
| [CLA INS P1 P2 Lc] ⇒ | |
| | ⇐ [INS] |
| [Data(Lc)] ⇒ | |
| | ⇐ 61 Licc |
| [00 C0 00 00 Licc] ⇒ | |
| | ⇐ C0 [Data(Licc)] 62 xx |

In either example, a R-APDU of {[Data(Licc)] 62 xx} is returned from the TTL to the TAL containing the data returned together with the warning status bytes.

***** END OF DOCUMENT *****