Answer to reset

From Wikipedia, the free encyclopedia
Jump to navigationJump to search

An **Answer To Reset** (**ATR**) is a message output by a contact Smart Card conforming to ISO/IEC 7816 standards, following electrical reset of the card's chip by a card reader. The ATR conveys information about the communication parameters proposed by the card, and the card's nature and state.

By extension, ATR often refers to a message obtained from a Smart Card in an early communication stage; or from the card reader used to access that card, which may transform the card's message into an ATR-like format (this occurs e.g. for some PC/SC card readers[1][2] when accessing an ISO/IEC 14443 Smart Card).

The presence of an ATR is often used as a first indication that a Smart Card appears operative, and its content examined as a first test that it is of the appropriate kind for a given usage.

Contact Smart Cards communicate over a signal named Input/Output (I/O) either *synchronously* (data bits are sent and received at the rhythm of one per period of the clock supplied to the card on its CLK signal) or *asynchronously* (data bits are exchanged over I/O with another mechanism for bit delimitation, similar to traditional asynchronous serial communication). The two modes are exclusive in a given communication session, and most cards are built with support for a single mode. Microprocessor-based contact Smart Cards are mostly of the asynchronous variety, used for all Subscriber Identity Modules (SIM) for mobile phones, those bank cards with contacts that conform to EMV specifications, all contact Java Cards, and Smart Cards for pay television. Memory-only cards are generally of the synchronous variety.

ATR under asynchronous and synchronous transmission have entirely different form and content. The ATR in asynchronous transmission is precisely normalized (in order to allow interoperability between cards and readers of different origin), and relatively complex to parse.

Some Smart Cards (mostly of the asynchronous variety) send different ATR depending on if the reset is the first since power-up (*Cold ATR*) or not (*Warm ATR*).

Note: Answer To Reset should not be confused with ATtRibute REQuest (ATR_REQ) and ATtRibute RESponse (ATR_RES) of NFC, also abbreviated ATR.[3] ATR_RES conveys information about the communication parameters supported, as does Answer To Reset, but its structure is different.

Contents

ATR in asynchronous transmission[edit]

The standard defining the ATR in asynchronous transmission is ISO/IEC 7816-3.[4] Subsets of the full ATR specification are used for some Smart Card applications, e.g. EMV.[5]

Physical form and timing at the card/reader interface[edit]

In asynchronous transmission, the ATR is transmitted by a card to a reader as characters, encoded as bits over the contact designated I/O (C7), with a nominal bit duration denoted Elementary Time Unit (ETU), equal during the whole ATR to 372

periods of the clock signal supplied by the reader on the CLK (C3) contact. The I/O line is by default at a H state (the highest voltage of two logic levels), and a transition to L state, denoted leading edge, defines the start of a character. The leading edge of the first character occurs between 400 and 40 000 clock cycles after the reader changed the RST (C2) contact from L to H.

Each characters comprises a start bit at L state, 8 data bits, 1 parity bit, followed (absent error) by a delay at the H state (a high voltage on I/O) such that the leading edge of characters in the ATR is at least 12 ETU, with amaximum designated Waiting Time WT = 9 600 ETU during the whole ATR (Eurocard MasterCard Visa specifications add that the reader should tolerate 10 800 ETU, that is 5% more). The value of the byte encoded by a character is defined according to conventions determined by the first character of the ATR, designated TS.

The end of the physical ATR between card and reader can be determined by the reader using analysis on the fly of the values of TS, T0, and any TD_i (see below), or/and on the basis of WT. The later method incurs an extra delay (about 0.8 s at the maximum clock frequency of 5 MHz applicable during ATR). EMV (but not ISO/IEC 7816-3) also allows the reader to consider that the ATR must be over after 20 160 ETU (about 1.5 s at 5 MHz) counted from the leading edge of TS.

Note: When communicating in asynchronous mode with an ISO/IEC 7816-3 contact Smart Card using a serial interface device operating per direct convention (such as a standard UART), it can be set to 8 bits, 1 even parity bit, 2 stop bits (sometime negotiable to 1, see TC_1); during the ATR, the baud rate should be 1/372 of the clock frequency received by the card (corresponding to an ETU of 372 clock cycles). There will normally be no parity error or framing error. The first byte received is `'3B'` if the card operates in direct convention, `'03'` if the cards operates in inverse convention, in which case the polarity and order of all 8 bits of each byte going through the serial interface device should be reversed, which in particular will change the first byte `'03'` to `'3F'`.

Historical note: provision for cards that use an internal clock source and a fixed ETU of 1/9 600 second during ATR existed in ISO/IEC 7816-3:1989, and was removed from the 1997 edition onwards.

General structure[edit]

The ATR proceeds in five steps: initial character TS; format byte T0; interface bytes TA_i, TB_i, TC_i, TD_i (optionals, variable number); historical bytes T_i (optionals, up to 15), and the check byte TCK (optional). There are a total of 2 to 33 characters including TS.

Name	Defines	Encodes	Present when
TS	Bit order and polarity		(always)
T0	Number of T_i, presence of $TA_1..TD_1$	K in [0..15]	(always)
TA_1	Maximum clock frequency, proposed bit duration	FI ↦ Fi and f_{max}; DI ↦ Di	5th bit of T0 is `1`
TB_1	Deprecated: V_{PP} requirements	PI1 ↦ P, II ↦ I	6th bit of T0 is `1`
TC_1	Extra delay between bytes required by card	N ↦ EGT ↦ GT	7th bit of T0 is `1`
TD_1	First offered transmission protocol, presence of $TA_2..TD_2$	T in [0..14]	8th bit of T0 is `1`
TA_2	Specific protocol and parameters to be used after the ATR	T in [0..14]	5th bit of TD_1 is `1`
TB_2	Deprecated: V_{PP} precise voltage requirement	PI2 ↦ P	6th bit of TD_1 is `1`
TC_2	Maximum waiting time for protocol T = 0	WI ↦ WT	7th bit of TD_1 is `1`
TD_2	A supported protocol or more global parameters, presence of $TA_3..TD_3$	T in [0..15]	8th bit of TD_1 is `1`
TA_i	For T = 1 [#]: maximum block size the card can receive If T = 15 [#]: supported supply voltages and low power modes	IFSC X; Y	5th bit of TD_{i-1} is `1`
TB_i	For T = 1 [#]: maximum delays between characters If T = 15 [#]: use of SPU contact C6	CWI ↦ CWT; BWI ↦ BWT	6th bit of TD_{i-1} is `1`
TC_i	For T = 1 [#]: type of error detection code used		7th bit of TD_{i-1} is `1`
TD_i	A supported protocol or more global parameters, presence of $TA_{i+1}..TD_{i+1}$	T in [0..15]	8th bit of TD_{i-1} is `1`
T_1	The format of historical bytes T_i		K ≥ 1
T_i	Historical bytes indicating operating characteristics, per ISO/IEC 7616-4 when T_1 is `'00'`, `'10'` or `'8X'`,		K ≥ *i*
TCK	Allow detection of accidental transmission error (the XOR of bytes T0 to TCK is normally zero)		T in any of the TD_i bytes is not 0

[#] The signification given is assuming $i > 2$, and i-1 is the only j with $1 < j < i$ such that TD_j encodes the stated value of T. When that T is in range [0..14], the signification of the byte applies only to the corresponding protocol (specific byte). When that T = 15, the signification applies regardless of protocol (global byte).

The initial character TS is always physically present, but is excluded of the Answer-to-Reset in the definition given by ISO/IEC 7816-3:2006: *the value of the byte string (at most 32 bytes) encoded in the sequence of characters following the initial character TS*. ISO/IEC 7816-4:2005 concurs,[6] stating that TS is a character or synchronization pattern, not a byte. However practice (in PC/SC, EMV, ETSI, and Calypso at least) is still to consider that TS is part of the ATR, as it was in ISO/IEC 7816-3:1997 and former. In particular, the ATR returned by PC/SC card readers and software stacks includes TS as the first byte, with value `'3B'` or `'3F'`.

Initial character TS[edit]

The initial character TS encodes the convention used for encoding of the ATR, and further communications until the next reset. In direct [resp. inverse] convention, bits with logic value `'1'` are transferred as a High voltage (H) [resp. a Low voltage (L)]; bits with logic value `'0'` are transferred as L [resp. H]; and least-significant bit of each data byte is first (resp. last) in the physical transmission by the card.

For direct convention, TS is `(H) L` *`H H L H H H L L`* `H (H)` and encodes the byte `'3B'`.

For inverse convention, TS is `(H) L` *`H H L L L L L L`* `H (H)` and encodes the byte `'3F'`.

[`(H)` represents the idle (High, Mark) state of the I/O line. The 8 data bits are shown in *italic*.]

Bits in bytes following TS in the ATR, and further communications until the next reset, are numbered 1st to 8th from low-order to high-order, and their value noted `0` or `1`, regardless of the chronological order and electrical representation, defined by TS. The bit following the 8 data bits in these bytes is an even parity bit, that is such that there's an even number of `'1'` bits (H or L according to the direct or inverse convention defined by TS) among the 8 data bits and the parity bit.

TS also allows the card reader to confirm or determine the ETU, as one third of the delay between the first and second H-to-L transition in TS. This is optional, and the principal definition of ETU in the ATR of standard-compliant asynchronous Smart Cards is 372 periods of the clock received by the card.

Format byte T0[edit]

The Format byte T0 encodes in its 4 low-order bits (4th MSbit to 1st LSbit) the number K of historical bytes T_i, in range [0..15].

It also encodes in its 4 high-order bits the presence of at most 4 other interface bytes: TA_1 (resp. TB_1, TC_1, TD_1) follow, in that order, if the 5th (resp. 6th, 7th, 8th) bit of T0 is `1`.

Interface bytes TA_i, TB_i, TC_i, TD_i[edit]

Interface bytes TA_1, TB_1, TC_1, TD_1, TA_2, TB_2, TC_2, TD_2, TA_3, TB_3, .. are all optional, and encode communication parameters and protocols that the card propose to use.

Interface bytes come in three kinds: *global* interface bytes apply to all protocols; *specific* interface bytes apply to a specific protocol; and *structural* interface bytes introduce further interface bytes, and protocols.

Interface byte TA_1[edit]

Interface byte TA_1, if present, is global, and encodes the maximum clock frequency f_{max} supported by the card, and the number of clock periods per ETU that it suggests to use after the ATR, expressed as the ratio Fi/Di of two integers. When TA_1 is absent, it's assumed default value is `'11'`, corresponding to f_{max} = 5 MHz, Fi = 372, Di = 1.

The 4 low-order bits of TA_1 (4th MSbit to 1st LSbit) encode Di as:

4th to 1st bits	`0000`	`0001`	`0010`	`0011`	`0100`	`0101`	`0110`	`0111`	`1000`	`1001`	`1010`	`1011`	`1100`	`1101`	`1110`	`1111`
Di	RFU	1	2	4	8	16	32	64(#)	12	20	RFU	RFU	RFU	RFU	RFU	RFU

(#) This was RFU in ISO/IEC 7816-3:1997 and former. Some card readers or drivers may erroneously reject cards using this value (or other RFU). Some PC/SC readers, as a workaround to said driver behavior, clear the 1st bit of TA_1 when its 4 low-order bits encode 7, and accordingly adjust TCK (if present), unless they have received a special command.

The 4 high-order bits of TA_1 (8th MSbit to 5th LSbit) encode f_{max} and Fi as:

8th to 5th bits	`0000`	`0001`	`0010`	`0011`	`0100`	`0101`	`0110`	`0111`	`1000`	`1001`	`1010`	`1011`	`1100`	`1101`	`1110`	`1111`
Fi	372(*)	372	558	744	1 116	1 488	1 860	RFU	RFU	512	768	1 024	1 536	2 048	RFU	RFU
f_{max} (MHz)	4(*)	5	6	8	12	16	20	—	—	5	7.5	10	15	20	—	—

(*) Historical note: in ISO/IEC 7816-3:1989, this was assigned to cards with internal clock, with no assigned Fi or f(max).

Note:EMV, and ISO/IEC 7816-3 before the 2006 edition, additionally use the notation DI (resp. FI) for the low-order (respectively high-order) 4 bits of TA_1. DI thus encodes Di, and FI encodes Fi and f_{max}.

Note: EMV's notation uses D (resp. F) where ISO/IEC 7816-3 uses Di (resp. Fi).

Example: TA_1 = `'B5'` = `10110101`, in which FI is `1011` and DI is `0101`, encodes f_{max} = 10 MHz, Fi = 1024, Di = 16, thus Fi/Di = 1024/16 = 64. This is inviting the card reader to take (after the ATR) the necessary steps to reduce the ETU to 64 clock cycles per ETU (from 372 during ATR) and increase the clock frequency up to 10 MHz (from perhaps 4 MHz during ATR).

Interface byte TB_1[edit]

TB_1, if present, is global. The usage of TB_1 is deprecated since the 2006 edition of the standard, which prescribes that cards *should* not include TB_1 in the ATR, and readers *shall* ignore TB_1 if present. EMV still requires that the card includes TB_1 = `'00'`, and that remains common practice; doing so explicitly indicates that the card does not use the dedicated contact C6 for the purpose of supplying a programming voltage (V_{PP}) to the card; the cards might however use C6 for Standard or Proprietary Use (SPU), such as communicating with a NFC front end by the Single Wire Protocol (SWP). On the reader side, EMV requires making a *warm* ATR for cards with TB_1 other than `'00'` in the *cold* ATR, and handling any TB_1 in a *warm* ATR as if it was `'00'`.

TB_1 was previously indicating (coarsely) the programming voltage V_{PP} and maximum programming current required by some cards on the dedicated contact C6 during programming of their EPROM memory. Modern Smart Cards internally generate the programming voltage for their EEPROM or Flash memory, and thus do not use V_{PP}. In the 1997 and earlier editions of the standard:

- The low 5 bits of TB_1 (5th MSbit to 1st LSbit) encode PI1; if TB_2 is absent, PI1 = 0 indicates that the C6 contact (assigned to V_{PP}) is not connected in the card; PI1 in range [5..25] encodes the value of V_{PP} in Volt (the reader shall apply that voltage only on specific demand by the card, with a tolerance of 2.5%, up to the maximum programming current; and otherwise leave the C6 contact used for V_{PP} within 5% of the V_{CC} voltage, up to 20 mA); if TB_2 is present, it supersedes the indication given by TB_1 in the PI1 field, regarding V_{PP} connection or voltage.

- The high bit of TB_1 (8th bits) is reserved, shall be `0`, and can be ignored by the reader.

- The 6th and 5th bits of TB_1 encode the maximum programming current (assuming neither TB_1 nor TB_2 indicate that V_{PP} is not connected in the card).

7th and 6th bits	`00`	`01`	`10`	`11`
Maximum programming current	25 mA	50 mA	RFU(#)	RFU

(#) This was 100 mA in ISO/IEC 7816-3:1989.

Interface byte TC_1[edit]

TC_1, if present, is global, and encodes the Extra Guard Time integer (N), from 0 to 255 (8th MSbit to 1st LSbit); otherwise, N = 0. N defines how much the Guard Time that the reader must apply varies from a baseline of 12 ETU (corresponding to 1 start bit, 8 data bits, 1 parity bit, and 2 stop bits; with the second stop bit possibly used for an error indication by the receiver under protocol T = 0). The Guard Time is the minimum delay between the leading edge of the previous character, and the leading edge of the next character sent.

Except when N is 255, the Guard Time is: GT = 12 ETU + R*N/f
where:
– f is the clock frequency being generated by the reader;
– R is some number of clock cycles, either:
  – per ETU, R = F/D, if T = 15 is absent from the ATR;
  – defined by TA_1, R = F_i/D_i (or its default value), if T = 15 is present in the ATR.

N = 255 has a protocol-dependent meaning: GT = 12 ETU during PPS (Protocol and Parameters Selection) and protocol T = 0, GT = 11 ETU under protocol T = 1 (corresponding to 1 start bit, 8 data bits, 1 parity bit, and 1 stop bit; with no error indication).

Except under protocol T = 1, the card transmits with a Guard Time of 12 ETU, irrespective of N. Under protocol T = 1, the Guard Time defined by N is also the Character Guard Time (CGT), and applies to card and reader for characters sent in the same direction.

Note: The reader remains bound by the Guard Time GT defined by N when other prescriptions specify another minimum delay between leading edges of characters in different directions, even when that minimum is lower than GT.

Historical note: ISO/IEC 7816-3:1989 only defined that N code the EGT as a number of ETU, the method now used when T = 15 is absent from the ATR. With this convention, cards that allow negotiation of a reduced number of clock cycles per ETU after PPS must also allow a proportionally reduced number of clock cycles for the EGT, which does not match with a common EGT motivation: account for delays before the card can receive the next character. The 1997 edition of the standard introduced that when T = 15 is present in the ATR, N code the EGT as a multiple of the number of clock cycles per ETU coded by TA_1, making the EGT effectively independent of the number of clocks cycles per ETU negotiated, while maintaining compatibility with former readers at least if they did not change the number of clock cycles per ETU.

Interface bytes TD_i[edit]

Interfaces bytes TD_i for $i \geq 1$, if present, are structural.

TD_i encodes in its 4 high-order bits the presence of at most 4 other interface bytes: TA_{i+1} (resp. TB_{i+1}, TC_{i+1}, TD_{i+1}) follow, in that order, if the 5th (resp. 6th, 7th, 8th) bit of TD_i is `1`.

TD_i encodes in its 4 low-order bits (4th MSbit to 1st LSbit) an integer T, in range [0..15]. T = 15 is invalid in TD_1, and in other TD_i qualifies the following TA_{i+1} TB_{i+1}, TC_{i+1}, TD_{i+1} (if present) as global interface bytes. Other values of T indicates a protocol that the card is willing to use, and that TA_{i+1} TB_{i+1}, TC_{i+1}, TD_{i+1} (if present) are specific interface bytes applying only to that protocol. T = 0 is a character-oriented protocol. T = 1 is a block-oriented protocol. T in the range [3..14] is RFU.

Historical note: provision for dynamically qualifying interface bytes as global using T = 15 did not exist in ISO/IEC 7816-3:1989.

Interface byte TA_2[edit]

Interface byte TA_2, if present, is global, and is named the *specific mode byte*.

Presence of TA_2 commands that the reader use *specific mode* as defined by TA_2 and earlier global bytes, rather than *negotiable mode* when TA_2 is absent.

TA_2 encodes in its 4 low-order bits an integer T defining the protocol required by the card, in the convention used for TD_1 (EMV prescribes that a card which T encoded in TA_2 does not match that in TD_1 shall be rejected).

The 5th bit is `0` to encodes that the required ETU duration is F_i/D_i clock cycles as defined by TA_1 (or its default value if absent); or `1` to indicate that the ETU duration is implicitly known (by some convention, or setting of the reader; EMV prescribes that such card shall be rejected).

The 6th and 7th bit are reserved for future use; `0` indicates not used.

The 8th bit is `1` to indicate that the card is unable to change the negotiable/specific mode (that is, does not propose other settings); or `0` to indicate that card has that ability (perhaps after a warm ATR).

Historical note: Provision for specific mode did not exist in ISO/IEC 7816-3:1989. Back then, the interface character TA_2 had no particular name or function, and was specific (to the protocol introduced by TD_1). ISO/IEC 7816-3:1997 introduced the specific mode and the specific mode byte, with interim note helping cards with specific mode byte TA_2 in their ATR dealing with a reader that did not implement specific mode.

Interface byte TB_2[edit]

TB_2, if present, is global. The usage of TB_2 is deprecated since the 2006 edition of the standard, which prescribes that cards *should* not include TB_2 in the ATR, and readers *shall* ignore TB_2 if present.

In the 1997 edition of the standard, TB_2 (8th to 1st bit) encode PI2, which when in range 50..250 (other values being RFU) encode V_{PP} in increments of 0.1 V, and subsumes the coarser indication given by PI1 of TB_1. Refer to that section for why modern Smart Cards have no use of V_{PP}, and thus of TB_2.

Historical note: Provision for TB_2 did not exist in ISO/IEC 7816-3:1989, and was introduced because V_{PP} = 12.5 V became a popular value in EEPROM technology, replacing 25 V and 21 V.

This section may **require cleanup** to meet Wikipedia's quality standards. The specific problem is: **Some of the ATR remains undocumented, including at least the meaning of TC_2, the first TA TB and TC for T = 15, and the interpretation of Historical bytes** Please help improve this section if you can. *(June 2014) (Learn how and when to remove this template message)*

Historical bytes T_i[edit]

Historical Characters T_i for $i \geq 1$, if present (as defined by K coded in T0), typically hold Information about the Card Builder, Type of Card (Size etc.), Version number and the State of the Card.

Check byte TCK[edit]

The ChecK byte (if present) allows a check of the integrity of the data in the ATR. If present, TCK is the Exclusive OR of the bytes in the ATR from T0 (included) to TCK (excluded).

TCK shall be present if and only if any of the TD_i present in the ATR encodes a value of T other than 0.

That rule for TCK presence is per ISO/IEC 7816-3:1989. The later ISO/IEC 7816-3:1997 and ISO/IEC 7816-3:2006 concur, at least whenever TA_2 is absent or encodes the same T as TD_1 (which is mandated by EMV). Common practice (e.g. in SIM cards) is to apply that rule, notwithstanding the contradictory prescription in EMV 4.3 Book 1, section 8.3.4, that *The ATR shall not contain TCK if T = 0 only is to be used*, instead reading that prescription as is if it ended in *if T = 0 only is indicated*.

ATR in synchronous transmission[edit]

The official reference defining the ATR in synchronous transmission is the ISO/IEC 7816-10 standard.[7]

The ATR starts with a header of 32 bits organized into 4 bytes, denoted H1 to H4. H1 codes the protocol (with `'00'` and `'FF'` being invalid), H2 codes parameters of the protocol. Little more is standardized.

References[edit]

1. ^ Section 5.3.3.1 in *SCM Microsystems SDI011 Reference Manual — version 1.05*
2. ^ Section 3.2 in *OMNIKEY Contactless Smart Card Readers Developer Guide* Archived October 6, 2011, at the Wayback Machine
3. ^ *ISO/IEC 18092:2004 — Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)*
4. ^ *ISO/IEC 7816-3:2006 — Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols* (partial preview)
5. ^ *[1]*, EMV 4.3 Integrated Circuit Card Specifications for Payment Systems — Book 1 — Application Independent ICC to Terminal Interface Requirements
6. ^ *[2]* (archived copy), ISO/IEC 7816-4:2005 (Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Organization, security and commands for interchange), note in section 7.4.2
7. ^ *ISO/IEC 7816-10:1999 — Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electronic signals and answer to reset for synchronous cards* (partial preview)

External links[edit]

- Smart card ATR parsing online ATR parsing tool

Retrieved from "https://en.wikipedia.org/w/index.php?title=Answer_to_reset&oldid=808475033"
Categories:

- Smart cards

Hidden categories:

- Webarchive template wayback links
- Articles needing cleanup from June 2014
- All pages needing cleanup
- Cleanup tagged articles with a reason field from June 2014
- Wikipedia pages needing cleanup from June 2014

Navigation menu

Personal tools

- Not logged in
- Talk
- Contributions
- Create account
- Log in

Namespaces

- Article

- Talk

Variants

Views

- Read
- Edit
- View history

More

Search

Search Wikipedia Go

Navigation

- Main page
- Contents
- Featured content
- Current events
- Random article
- Donate to Wikipedia
- Wikipedia store

Interaction

- Help
- About Wikipedia
- Community portal
- Recent changes
- Contact page

Tools

- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Page information
- Wikidata item
- Cite this page

Print/export

- Create a book
- Download as PDF
- Printable version

Languages

- Deutsch
- Bahasa Melayu

Edit links

- This page was last edited on 3 November 2017, at 02:30 (UTC).