



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR122U

Čtečka USB NFC



Rozhraní pro programování aplikací V2.04

Obsah

1.0.	Úvod	4
1.1.	Funkce	4
1.2.	Rozhraní USB	5
2.0.	Provádění	6
2.1.	Komunikační schéma ACR122U	6
2.2.	Přehled rozhraní čtečky čipových karet	7
3.0.	Rozhraní PICC Popis	8
3.1.	Generování ATR	8
3.1.1.	Formát ATR pro PICC podle ISO 14443 část 3	8
3.1.2.	Formát ATR pro PICC podle ISO 14443 část 4	9
4.0.	Příkazy PICC pro obecné účely	11
4.1.	Získat data	11
5.0.	Příkazy PICC (emulace T=CL) pro paměťové karty MIFARE Classic	12
5.1.	Načtení ověřovacích klíčů	12
5.2.	Ověřování	13
5.3.	Čtení binárních bloků	16
5.4.	Aktualizace binárních bloků	17
5.5.	Související příkazy bloku hodnot	18
5.5.1.	Operace bloku hodnot	18
5.5.2.	Blok čtení hodnot	19
5.5.3.	Blok obnovení hodnoty	20
6.0.	Příkazy pseudoAPDU	21
6.1.	Přímý přenos	21
6.2.	Dvoubarevná LED dioda a bzučák	22
6.3.	Zjištění verze firmwaru čtečky	24
6.4.	Získejte provozní parametr PICC	25
6.5.	Nastavení provozního parametru PICC	26
6.6.	Nastavení parametru časového limitu	27
6.7.	Nastavení výstupu bzučáku při detekci karty	28
7.0.	Základní tok programu pro bezkontaktní aplikace	29
7.1.	Jak přistupovat ke značkám kompatibilním s PC/SC (ISO 14443-4)?	31
7.2.	Jak přistupovat k tagům MIFARE DESFire (ISO 14443-4)?	32
7.3.	Jak získat přístup ke značkám FeliCa (ISO 18092)?	34
7.4.	Jak přistupovat ke štítkům NFC fóra typu 1 (ISO 18092)?	35
7.5.	Získání aktuálního nastavení bezkontaktního rozhraní	37
Dodatek A.	ACR122U Příkaz PC/SC Escape	38
Dodatek B.	Tok příkazů a odpovědí APDU pro značky kompatibilní s ISO 14443	41
Dodatek C.	Tok příkazů a odpovědí APDU pro značky kompatibilní s ISO 18092	42
Dodatek D.	Chybové kódy	43
Příloha E.	Vzorové kódy pro nastavení LED diody	45

Seznam obrázků

Obrázek 1	: Komunikační schéma ACR122U	6
Obrázek 2	: Rozhraní čtečky čipových karet ve Správci zařízení	7



Obrázek 3 : Základní tok programu pro bezkontaktní aplikace29
Obrázek 4 : Mapa paměti Topaz36

Seznam tabulek

Tabulka 1 : Rozhraní USB.....5
Tabulka 2 : Formát ATR pro PICC podle ISO 14443 část 3.....8
Tabulka 3 : Formát ATR pro PICC podle ISO 14443 část 4.....9
Tabulka 4 : Mapa paměti MIFARE 1K.....14
Tabulka 5 : Mapa paměti MIFARE Classic 4K14
Tabulka 6 : Mapa paměti MIFARE Ultralight15
Tabulka 7 : Chybové kódy.....44



1.0. Úvod

ACR122U je bezkontaktní čtečka/zapisovač čipových karet připojená k počítači, která se používá pro přístup ke značkám ISO 14443-4 typu A a B, MIFARE®, ISO 18092 a FeliCa. ACR122U je kompatibilní s PC/SC, takže je kompatibilní se stávajícími aplikacemi PC/SC.

ACR122U slouží jako prostředník mezi počítačem a bezkontaktním štítkem prostřednictvím rozhraní USB. Čtečka provede příkaz z počítače, ať už je příkaz použit ke komunikaci s bezkontaktním tagem, nebo k ovládání periférií zařízení (LED nebo bzučák). Tento dokument API se podrobně zabývá tím, jak byly implementovány příkazy PC/SC pro bezkontaktní rozhraní a periferie zařízení ACR122U.

1.1. Funkce

- Rozhraní USB 2.0 s plnou rychlostí
- Soulad s CCID
- Čtečka čipových karet:
 - Rychlost čtení/zápisu až 424 Kb/s
 - Vestavěná anténa pro bezkontaktní přístup k tagům se vzdáleností čtení karet až 50 mm (v závislosti na typu tagu).
 - Podpora karet ISO 14443 část 4 typ A a B, MIFARE, FeliCa a všech čtyř typů NFC (tagy ISO/IEC 18092).
 - Vestavěná funkce proti kolizi (v jednom okamžiku je přístupný pouze jeden štítek)
- Rozhraní pro programování aplikací:
 - Podpora PC/SC
 - Podpora rozhraní CT-API (prostřednictvím wrapperu nad PC/SC)
- Vestavěné periferie:
 - Uživatelsky ovladatelná dvoubarevná LED dioda
 - Uživatelem ovládaný bzučák
- Podporuje systém Android™ 3.1 a vyšší
- Splňuje následující normy:
 - IEC/EN 60950
 - ISO 18092
 - ISO 14443
 - CE
 - FCC
 - KC
 - VCCI
 - MIC
 - PC/SC
 - CCID
 - Microsoft® WHQL
 - RoHS 2



1.2. Rozhraní USB

ACR122U se připojuje k počítači prostřednictvím rozhraní USB podle specifikace USB 1.1. ACR122U pracuje v režimu plné rychlosti, tj. 12 Mb/s.

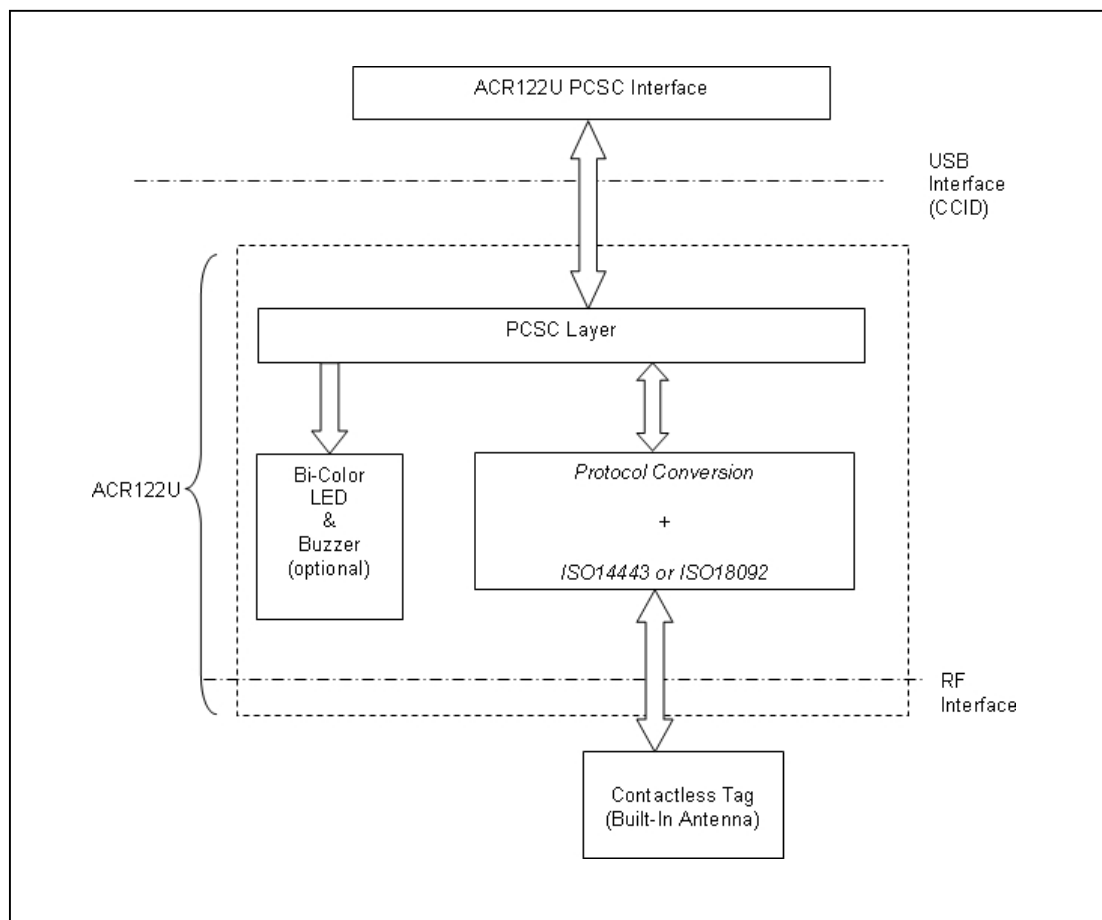
Kolík	Signál	Funkce
1	V _{BUS}	Napájení +5 V pro čtečku (max. 200 mA, normální 100 mA)
2	D-	Diferenciální signál přenáší data mezi ACR122U a PC
3	D+	Diferenciální signál přenáší data mezi ACR122U a PC
4	GND	Referenční úroveň napětí pro napájení

Tabulka 1: Rozhraní USB

2.0. Provádění

2.1. Komunikační schéma ACR122U

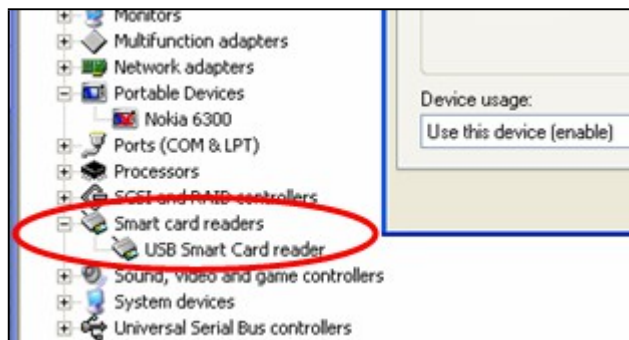
Používají se standardní ovladače Microsoft CCID a PC/SC; nejsou tedy nutné žádné ovladače ACS, protože ovladače jsou již zabudovány v operačním systému Windows®. Nastavení registru počítače lze také upravit tak, aby bylo možné plně využívat možnosti čtečky ACR122U NFC. Další podrobnosti naleznete v [dodatku A](#).



Obrázek 1: Komunikační schéma ACR122U

2.2. Přehled rozhraní čtečky čipových karet

Přejděte do **Správce zařízení** a zobrazte "ACR122U PICC Interface". Měl by být použit standardní ovladač Microsoft USB CCID Driver.



Obrázek 2: Rozhraní čtečky čipových karet ve Správci zařízení

3.0. Rozhraní PICC Popis

3.1. Generování ATR

Pokud čtečka detekuje PICC, odešle se ovladači PC/SC zpráva ATR pro identifikaci PICC.

3.1.1. Formát ATR pro PICC podle ISO 14443 část 3

Byte	Hodnota (Hex)	Označení	Popis
0	3Bh	Úvodní záhlaví	-
1	<u>8Nh</u>	T0	Vyšší nibble 8 znamená: žádné TA1, TB1, TC1, následuje pouze TD1. Dolní nibble N je počet historických bajtů (HistByte 0 až HistByte N-1).
2	80h	TD1	Vyšší nibble 8 znamená: žádné TA2, TB2, TC2, následuje pouze TD2. Dolní nibble 0 znamená T = 0
3	01h	TD2	Vyšší nibble 0 znamená, že nenásleduje TA3, TB3, TC3, TD3. Dolní nibble 1 znamená T = 1
4 Na 3+N	80h	T1	Byte indikátoru kategorie, 80 znamená Indikátor stavu může být přítomen ve volitelném datovém objektu COMPACT-TLV.
	4Fh	Tk	Indikátor aplikace Indikátor přítomnosti
	0Ch		Délka
	RID		Registrovaný identifikátor poskytovatele aplikace (RID) # A0 00 00 03 06h
	SS		Byte pro standard
	C0 .. C1h		Bajty pro název karty
	00 00 00 00h		RFU
4+N	UUh	TCK	Exkluzivní řazení všech bajtů T0 až Tk

Tabulka 2: Formát ATR pro PICC podle ISO 14443 část 3

Příklad:

ATR pro MIFARE 1K = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6Ah}

ATR											
Úvodní záhlaví	T0	TD1	TD2	T1	Tk	Délka	RID	Standardní	Název v karty	RFU	TCK
3Bh	8Fh	80h	01h	80h	4Fh	0Ch	A0 00 00 03 06h	03h	00 01h	00 00 00 00h	6Ah



Kde:

Délka (YY) = 0Ch

RID = A0 00 00 03 06h (pracovní skupina PC/SC)

Standardní (SS) = 03h (ISO 14443A, část 3)

Název karty (C0 .. C1) = [00 01h] (MIFARE Classic® 1K)

Kde, Název karty (C0 .. C1)

00 01h: 00 02h: MIFARE

Classic 4K 00 03h: MIFARE®

Ultralight® 00 26h: MIFARE

Mini

F0 04h: F0 11h: FeliCa

212K

F0 12h: FeliCa 424K

FFh [SAK]: Neurčeno

3.1.2. Formát ATR pro PICC podle ISO 14443 část 4

Byte	Hodnota (Hex)	Označení	Popis
0	3Bh	Úvodní záhlaví	-
1	<u>8Nh</u>	T0	Vyšší nibble 8 znamená: žádné TA1, TB1, TC1, následuje pouze TD1. Dolní nibble N je počet historických bajtů (HistByte 0 až HistByte N-1).
2	80h	TD1	Vyšší nibble 8 znamená: žádné TA2, TB2, TC2, následuje pouze TD2. Dolní nibble 0 znamená T = 0
3	01h	TD2	Vyšší nibble 0 znamená, že nenásleduje TA3, TB3, TC3, TD3. Dolní nibble 1 znamená T = 1
4 na 3 + N	XXh	T1	Historické byty:
	XXh XX XXh	Tk	ISO 14443A: Historické bajty z odpovědi ATS. Viz specifikace ISO14443-4. ISO 14443B: Odpověď vyšší vrstvy z odpovědi ATTRIB (ATQB). Viz specifikace ISO14443-3.
4+N	UUh	TCK	Exkluzivní řazení všech bajtů T0 až Tk

Tabulka 3: Formát ATR pro PICC podle ISO 14443 část 4



Vezměme si například ATR pro DESFire, což je:
DESFire (ATR) = 3B 86 80 01 06 75 77 81 02 80 00h

ATR						
Úvodní záhlaví	T0	TD1	TD2	ATS		TCK
				T1	Tk	
3Bh	86h	80h	01h	06h	75 77 81 02 80h	00h

Tento ATR má 6 bajtů ATS, což je: [06 75 77 81 02 80h]

Poznámka: Pro rozlišení PICC ISO 14443A-4 a ISO 14443B-4 použijte APDU "FF CA 01 00 00h" a načtěte úplný ATS, je-li k dispozici. ATS se vrací pro PICC ISO14443A-3 nebo ISO14443B-3/4.

Dalším příkladem může být ATR pro ST19XRC8E, který je:

ST19XRC8E (ATR) = 3B 8C 80 01 50 12 23 45 56 12 53 54 4E 33 81 C3 55h

ATR						
Úvodní záhlaví	T0	TD1	TD2	ATQB		TCK
				T1	Tk	
3Bh	86h	80h	01h	50h	12 23 45 56 12 53 54 4E 33 81 C3h	55h

Protože tato karta odpovídá normě ISO 14443 typu B, odpověď by byla ATQB, což je 50 12 23 45 56 12. 53 54 4E 33 81 C3h je 12 bajtů dlouhý bez CRC-B

Poznámka: Další podrobnosti naleznete v normách ISO7816, ISO14443 a PC/SC.

4.0. Příkazy PICC pro obecné účely

4.1. Získat data

Tento příkaz vrací sériové číslo nebo ATS připojeného PICC. Get UID
Formát APDU (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Získat data	FFh	CAh	00h 01h	00h	00h (plná délka)

Získat formát odpovědi UID (UID + 2 bajty), pokud P1 = 00h

Reakce	Data Out					
Výsledek	UID (LSB)	-	-	UID (MSB)	SW1	SW2

Získat ATS karty ISO 14443 A (ATS + 2 bajty), pokud P1 = 01h

Reakce	Data Out		
Výsledek	ATS	SW1	SW2

Kódy odpovědi

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.
Chyba	6A 81h	Funkce není podporována.

Příklad:

- Zjištění sériového čísla připojeného PICC.
UINT8 GET_UID[5]={FFh, CAh, 00h, 00h, 04h};
- Získání ATS připojeného PICC ISO 14443 A. UINT8
GET_ATS[5]={FFh, CAh, 01h, 00h, 04h};

5.0. Příkazy PICC (emulace T=CL) pro paměťové karty MIFARE Classic

5.1. Načtení ověřovacích klíčů

Tento příkaz načte do čtečky ověřovací klíče. Autentizační klíče slouží k ověření konkrétního sektoru paměťové karty MIFARE Classic 1K/4K. K dispozici je volatelné umístění autentizačních klíčů.

Formát APDU pro načítání ověřovacích klíčů (11 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data
Načtení ověřovacích klíčů	FFh	82h	Klíčová struktura	Číslo klíče	06h	Klíč (6 bajtů)

Kde:

Struktura klíče 1 bajt.

00h = Klíč je nahrán do volatelné paměti čtečky. Ostatní = vyhrazeno.

Číslo klíče 1 bajt.

00h ~ 01h = umístění klíče. Klávesy zmizí, jakmile je čtečka odpojována od počítače.

Klíč 6 bajtů.

Hodnota klíče načtená do čtečky. např. {FF FF FF FF FF FF FFh}.

Načíst ověřovací klíče Formát odpovědi (2 bajty)

Reakce	Data Out	
Výsledek	SW1	SW2

Kódy odpovědi

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

Příklad:

Nahrajte klíč {FF FF FF FF FF FF FFh} do umístění klíče 00h.

APDU = {FF 82 00 00h 06 FF FF FF FF FF FF FFh}

5.2. Ověřování

Tento příkaz používá klíče uložené ve čtečce k ověření pomocí karty MIFARE 1K/4K (PICC). Používají se dva typy autentizačních klíčů: TYPE_A a TYPE_B.

Formát APDU pro načítání ověřovacích klíčů (6 bajtů) [Zastaralé]

Příkaz	Třída	INS	P1	P2	P3	Vstupní data
Ověřování	FFh	88h	00h	Číslo bloku	Typ klíče	Číslo klíče

Formát APDU pro načítání ověřovacích klíčů (10 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data
Ověřování	FFh	86h	00h	00h	05h	Ověřování datových bytů

Autentizační datové bajty (5 bajtů)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Verze 01h	00h	Číslo bloku	Typ klíče	Číslo klíče

Kde:

Číslo bloku 1 bajt. Jedná se o blok paměti, který má být ověřen.

Typ klíče 1 bajt

60h = Klíč se používá jako klíč TYPU A pro ověřování.

61h = Klíč se používá jako klíč TYPU B pro ověřování.

Číslo klíče 1 bajt

00h ~ 01h = umístění klíče.

Poznámka: Karta MIFARE Classic 1K má celkem 16 sektorů a každý sektor se skládá ze 4 po sobě jdoucích bloků. Např. sektor 00h se skládá z bloků {00h, 01h, 02h a 03h}; sektor 01h se skládá z bloků {04h, 05h, 06h a 07h}; poslední sektor 0F se skládá z bloků {3Ch, 3Dh, 3Eh a 3Fh}.

Po úspěšném ověření není třeba ověření opakovat, pokud bloky, ke kterým se má přistupovat, patří do stejného sektoru. Další podrobnosti naleznete ve specifikaci MIFARE Classic 1K/4K.

Formát odpovědi na otázku Load Authentication Keys (2 bajty)

Reakce	Data Out	
Výsledek	SW1	SW2

Kódy odpovědi

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

Odvětví (Celkem 16 sektorů. Každý sektor se skládá ze 4 po sobě jdoucích bloků)	Datové bloky (3 bloky, 16 bajtů na blok)	Blok přívěsů (1 blok, 16 bajtů)
Sektor 0	00h ~ 02h	03h
Sektor 1	04h ~ 06h	07h
..		
..		
Sektor 14	38h ~ 0Ah	3Bh
Sektor 15	3Ch ~ 3Eh	3Fh

} 1 KB

Tabulka 4: Mapa paměti MIFARE 1K

Odvětví (Celkem 32 sektorů. Každý sektor se skládá ze 4 po sobě jdoucích bloků)	Datové bloky (3 bloky, 16 bajtů na blok)	Blok přívěsů (1 blok, 16 bajtů)
Sektor 0	00h ~ 02h	03h
Sektor 1	04h ~ 06h	07h
..		
..		
Sektor 30	78h ~ 7Ah	7Bh
Sektor 31	7Ch ~ 7Eh	7Fh

} 2 KB

Odvětví (Celkem 8 sektorů. Každý sektor se skládá z 16 po sobě jdoucích bloků)	Datové bloky (15 bloků, 16 bajtů na blok)	Blok přívěsů (1 blok, 16 bajtů)
Sektor 32	80h ~ 8Eh	8Fh
Sektor 33	90h ~ 9Eh	9Fh
..		
..		
Sektor 38	E0h ~ EEh	EFh
Sektor 39	F0h ~ FEh	FFh

} 2 KB

Tabulka 5: Mapa paměti MIFARE Classic 4K

Číslo bajtu	0	1	2	3	Stránka
Sériové číslo	SN0	SN1	SN2	BCC0	0
Sériové číslo	SN3	SN4	SN5	SN6	1
Interní/zámek	BCC1	Interní	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	3
Čtení/zápis dat	Data0	Data1	Data2	Data3	4
Čtení/zápis dat	Data4	Data5	Data6	Data7	5
Čtení/zápis dat	Data8	Data9	Data10	Data11	6
Čtení/zápis dat	Data12	Data13	Data14	Data15	7
Čtení/zápis dat	Data16	Data17	Data18	Data19	8
Čtení/zápis dat	Data20	Data21	Data22	Data23	9
Čtení/zápis dat	Data24	Data25	Data26	Data27	10
Čtení/zápis dat	Data28	Data29	Data30	Data31	11
Čtení/zápis dat	Data32	Data33	Data34	Data35	12
Čtení/zápis dat	Data36	Data37	Data38	Data39	13
Čtení/zápis dat	Data40	Data41	Data42	Data43	14
Čtení/zápis dat	Data44	Data45	Data46	Data47	15

512
bitů
nebo
64 bajtů

Tabulka 6: Mapa paměti MIFARE Ultralight

Příklad:

- Ověření bloku 04h pomocí {TYP A, číslo klíče 00h}. Pro PC/SC V2.01, zastaralé. APDU = {FF 88 00 04 60 00h};
- Ověření bloku 04h pomocí {TYP A, číslo klíče 00h}. Pro PC/SC V2.07 APDU = {FF 86 00 00 05 01 00 04 60 00h}.

Poznámka: MIFARE Ultralight nepotřebuje provádět žádné ověřování. Přístup do paměti je volný.

5.3. Čtení binárních bloků

Tento příkaz načte datové bloky z PICC. Datový blok/blok návěsu musí být nejprve ověřen.

Čtení binárního formátu APDU (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Čtení binárních bloků	FFh	B0h	00h	Číslo bloku	Počet bajtů ke čtení

Kde:

Číslo bloku	1 byte
	Blok, ke kterému se má přistupovat.
Počet bajtů ke čtení	1 bajt
	Maximálně 16 bajtů.

Formát odpovědi při čtení binárního bloku (N + 2 bajty)

Reakce	Data Out		
Výsledek	0 ≤ N ≤ 16	SW1	SW2

Kódy odpovědi

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

Příklad:

- Čtení 16 bajtů z binárního bloku 04h (MIFARE Classic 1K nebo 4K)
APDU = {FF B0 00 04 10h}
- Čtení 4 bajtů z binární stránky 04h (MIFARE Ultralight)
APDU = {FF B0 00 04h}
- Čtení 16 bajtů počínaje binární stránkou 04h (MIFARE Ultralight) (čtou se stránky 4, 5, 6 a 7).
APDU = {FF B0 00 04 10h}

Poznámka: Při čtení zpráv NDEF na kartách MIFARE Classic 4K počítejte se zpožděním 2 sekundy.

5.4. Aktualizace binárních bloků

Tento příkaz zapisuje datové bloky do PICC. Datový blok/blok návěsu musí být ověřen.

Aktualizace binárního formátu APDU (4 nebo 16 + 5 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data
Aktualizace binárních bloků	FFh	D6h	00h	Číslo bloku	Počet bajtů k aktualizaci	Bloková data 4 bajty pro MIFARE Ultralight nebo 16 bajtů pro MIFARE 1K/4K

Kde:

Číslo bloku	1 bajt
	Výchozí blok, který má být aktualizován.
Počet bajtů k aktualizaci	1 bajt
	16 bajtů pro MIFARE 1K/4K
	4 bajty pro MIFARE Ultralight
Bloková data	4 nebo 16 bajtů.
	Data, která mají být zapsána do binárního bloku/bloků.

Kódy odpovědí

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

Příklad:

- Aktualizace binárního bloku 04h MIFARE Classic 1K/4K s daty {00 01 .. 0Fh}
APDU = {FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0Fh}
- Aktualizace binárního bloku 04h MIFARE Ultralight s daty {00 01 02 03}.
APDU = {FF D6 00 04 04 00 01 02 03h}

5.5. Související příkazy bloku hodnot

Datový blok lze použít jako hodnotový blok pro implementaci aplikací založených na hodnotách.

5.5.1. Operace bloku hodnot

Tento příkaz manipuluje s transakcemi založenými na hodnotách (např. zvyšuje hodnotu bloku hodnot apod.).

Operace bloku hodnot Formát APDU (10 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data	
Operace bloku hodnot	FFh	D7h	00h	Číslo bloku	05h	VB_OP	VB_Value (4 bajty) {MSB .. LSB}

Kde:

Číslo bloku	1 byte	Blok hodnot, se kterým se má manipulovat.
VB_OP	1 byte	00h = Uložení hodnoty VB_Value do bloku. Blok bude poté převeden na blok hodnot. 01h = Zvýšení hodnoty bloku hodnot o VB_Value. Tento příkaz je platný pouze pro blok hodnot. 02h = Snížení hodnoty bloku hodnot o VB_Value. Tento příkaz je platný pouze pro blok hodnot.
VB_Value	4 bajty.	Hodnota použitá pro manipulaci s hodnotou. Hodnota je dlouhé celé číslo se znaménkem (4 bajty).

Příklad 1: = {FFh, FFh, FFh, FCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

Příklad 2: Desetinné číslo 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

Formát odpovědi na operaci bloku hodnot (2 bajty)

Reakce	Data Out	
Výsledek	SW1	SW2

Kódy odpovědí

Výsledek	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

5.5.2. Blok čtení hodnot

Tento příkaz načte hodnotu z bloku hodnot. Tento příkaz je platný pouze pro blok hodnot.

Formát bloku čtené hodnoty APDU (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Blok čtení hodnot	FFh	B1h	00h	Číslo bloku	04h

Kde:

Číslo bloku 1 byte

Blok hodnot, ke kterému se má přistupovat.

Formát odpovědi bloku čtené hodnoty (4 + 2 bajty)

Reakce	Data Out		
Výsledek	Hodnota {MSB .. LSB}	SW1	SW2

Kde:

Hodnota 4 bajty.

Hodnota vrácená z karty. Hodnota je dlouhé celé číslo se znaménkem (4 bajty).

Příklad 1: = {FFh, FFh, FFh, FCh}

Hodnota			
MSB			LSB
FFh	FFh	FFh	FCh

Příklad 2: Desetinné číslo 1 = {00h, 00h, 00h, 01h}

Hodnota			
MSB			LSB
00h	00h	00h	01h

Kódy odpovědí

Výsledek	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

5.5.3. Blok obnovení hodnoty

Tento příkaz zkopíruje hodnotu z bloku hodnot do jiného bloku hodnot.

Formát bloku APDU pro obnovení hodnoty (7 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data	
Blok obnovení hodnoty	FFh	D7h	00h	Číslo zdrojového bloku	02h	03h	Číslo cílového bloku

Kde:

Číslo zdrojového bloku 1 byte

Hodnota zdrojového bloku hodnot se zkopíruje do cílového bloku hodnot.

Číslo cílového bloku 1 byte

Blok hodnot, který má být obnoven. Zdrojový a cílový blok hodnot musí být ve stejném sektoru.

Formát odpovědi bloku obnovení hodnoty (2 bajty)

Reakce	Data Out	
Výsledek	SW1	SW2

Kódy odpovědi

Výsledek	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

Příklad:

- Uložení hodnoty "1" do bloku 05h
APDU = {FF D7 00 05 05 00 00 00 00 01h}
Odpověď: 90 00h
- Čtení bloku hodnot 05h
APDU = {FF B1 00 05 00h}
Odpověď: 00 00 00 01 90 00h [9000h]
- Kopírování hodnoty z bloku hodnot 05h do bloku hodnot 06h APDU = {FF D7 00 05 02 03 06h}
Odpověď: 90 00h [9000h]
- Zvýšení bloku hodnot 05h o "5" APDU =
{FF D7 00 05 05 01 00 00 00 05h}
Odpověď: 90 00h [9000h]

6.0. Příkazy pseudoAPDU

Příkazy pseudoAPDU se používají k následujícím účelům:

- Výměna dat se značkami, které nejsou kompatibilní s PC/SC
- Získání a nastavení parametrů čtečky
- Pseudo-APDU lze odesílat přes "ACR122U PICC Interface", pokud je tag již připojen.
- Pseudo-APDU lze odeslat pomocí příkazu "Escape Command", pokud značka ještě není prezentována.

6.1. Přímý přenos

Jedná se o užitečné zatížení, které má být odesláno do tagu nebo čtečky.

Formát příkazu pro přímý přenos (délka užitečného zatížení + 5 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data
Přímý přenos	FFh	00h	00h	00h	Počet odesílaných bajtů	Užitečné zatížení

Kde:

Lc 1 bajt.
 Počet odesílaných bajtů
 Maximálně 255 bajtů

Vstupní data Reakce

Formát odpovědi na přímý přenos

Reakce	Data Out
Přímý přenos	Údaje o odezvě

6.2. Dvoubarevná LED dioda a bzučák

Tento příkaz ovládá stavy dvoubarevné LED a bzučáku.

Formát příkazu pro ovládání dvoubarevné LED a bzučáku (9 bajtů)

Příkaz	Třída	INS	P1	P2	Lc	Data In (4 bajty)
Dvoubarevný a bzučák Ovládání LED	FFh	00h	40h	LED Řízení stavu (bit 7 --- bit 0)	04h	Řízení doby trvání blikání

Kde:

P2 Řízení stavu LED

Formát ovládání dvoubarevné LED a bzučáku (1 byte)

CMD	Položka	Popis
Bit 0	Konečný stav: Červená LED dioda	1 = zapnuto; 0 = vypnuto
Bit 1	Konečný stav: Zelená LED dioda	1 = zapnuto; 0 = vypnuto
Bit 2	Státní maska: Červená LED dioda	1 = Aktualizovat stav 0 = Beze změny
Bit 3	Státní maska: Zelená LED dioda	1 = Aktualizovat stav 0 = Beze změny
Bit 4	Počáteční stav blikání: Červená LED dioda	1 = zapnuto; 0 = vypnuto
Bit 5	Počáteční stav blikání: Zelená LED dioda	1 = zapnuto; 0 = vypnuto
Bit 6	Blikající maska: Červená LED dioda	1 = blikat 0 = neblíká
Bit 7	Blikající maska: Zelená LED dioda	1 = blikat 0 = neblíká

Vstupní data Řízení doby trvání blikání

Dvoubarevné blikání LED Doba trvání Kontrolní formát (4 byty)

Byte 0	Byte 1	Byte 2	Byte 3
T1 Doba trvání Počáteční stav blikání (Jednotka = 100 ms)	T2 Trvání Přepínání stavu blikání (Jednotka = 100 ms)	Počet opakování	Odkaz na bzučák

Kde:

Byte 3 Odkaz na bzučák. Ovládání stavu bzučáku během blikání LED.

00h: Bzučák se nezapne

01h: 02h: bzučák se zapne během doby trvání T1:

Bzučák se zapne během doby trvání T2

03h: Během doby trvání T1 a T2 se zapne bzučák.



Výsledky	SW1	SW2	Význam
Úspěch	90h	Aktuální stav LED	Operace byla úspěšně dokončena.
Chyba	63h	00h	Operace se nezdařila.

Aktuální stav LED (1 byte)

Stav	Položka	Popis
Bit 0	Aktuální červená LED dioda	1 = zapnuto; 0 = vypnuto
Bit 1	Aktuální zelená LED dioda	1 = zapnuto; 0 = vypnuto
Bity 2 - 7	Rezervováno	

Poznámky:

1. *Blikání LED se projeví pouze v případě, že je povolena příslušná maska blikání LED a počet opakování je větší než nula.*
2. *Termín počáteční stav blikání znamená, že LED zvolené barvy se během prvního bliknutí v pracovním cyklu buď rozsvítí, nebo zhasne. Například pokud je Initial Blinking State (Počáteční stav blikání) zapnut pro zelenou LED a vypnut pro červenou LED, pak blikání začne zelenou barvou, následovat bude červená atd.*
3. *Změna stavu LED se projeví pouze v případě, že je povolena odpovídající maska stavu LED.*
4. *Pokud je ovládáno současně, operace Stav LED se provede po dokončení operace Blikání LED.*
5. *V části Řízení doby trvání blikání se pro řízení pracovního cyklu blikání LED a doby zapnutí bzučáku používají parametry doby trvání T1 a T2. Například pokud T1=1 a T2=1, pracovní cyklus = 50%. #Duty Cycle = T1/(T1 + T2).*
6. *Chcete-li ovládat pouze bzučák, nastavte P2 "LED State Control" na nulu.*
7. *Aby bzučák fungoval, musí být "počet opakování" větší než nula.*
8. *Chcete-li ovládat pouze LED, nastavte parametr "Link to Buzzer" na nulu.*



6.3. Zjištění verze firmwaru čtečky

Tento příkaz zjistí verzi firmwaru čtečky.

Formát příkazu (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Získat verzi firmwaru	FFh	00h	48h	00h	00h

Formát odpovědi (10 bajtů)

Reakce	Data Out
Výsledek	Verze firmwaru

Příklad:

Odpověď = 41 43 52 31 32 32 55 32 30 31h = ACR122U201 (ASCII)



6.4. Získejte provozní parametr PICC

Tento příkaz načte provozní parametr PICC čtečky.

Formát příkazu (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Získat provozní parametr PICC	FFh	00h	50h	00h	00h

Formát odpovědi (2 bajty)

Reakce	Data Out	
Výsledek	90h	Provozní parametr PICC

6.5. Nastavení provozního parametru PICC

Tento příkaz nastavuje provozní parametr PICC čtečky.

Formát příkazu (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Nastavení PICC Provozní parametr	FFh	00h	51h	Nový provozní parametr PICC	00h

Formát odpovědi (2 bajty)

Reakce	Data Out	
Výsledek	90h	Provozní parametr PICC

Provozní parametr PICC

Bit	Parametr	Popis	Možnost
7	Automatické dotazování PICC	Povolení funkce PICC Polling	1 = Povolit 0 = Zakázat
6	Automatické generování ATS	Vydání požadavku ATS vždy, když je aktivována značka ISO14443-4 typu A.	1 = Povolit 0 = Zakázat
5	Interval dotazování	Nastavení časového intervalu mezi po sobě jdoucími PICC Polling.	1 = 250 ms 0 = 500 ms
4	FeliCa 424K	Typy značek, které mají být detekovány během dotazování PICC.	1 = detekovat 0 = Přeskočit
3	FeliCa 212K		1 = detekovat 0 = Přeskočit
2	Topaz		1 = detekovat 0 = Přeskočit
1	ISO 14443 typ B		1 = detekovat 0 = Přeskočit
0	ISO 14443 typ A # Aby bylo možné detekovat štítky MIFARE, musí být nejprve vypnuto automatické generování ATS.		1 = detekovat 0 = Přeskočit

Poznámka: Výchozí hodnota = FFh

6.6. Nastavení parametru časového limitu

Tento příkaz nastavuje parametr timeout doby odezvy bezkontaktního čipu.

Formát příkazu (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Nastavení parametru časového limitu	FFh	00h	41h	Parametr časového limitu (jednotka: 5 s)	00h

Kde:

P2 Parametr časového limitu

00h: Žádná kontrola časového limitu

01h - FEh: Timeout s jednotkou 5 sekund

FFh: Počkejte, dokud bezkontaktní čip neodpoví.

Formát odpovědi (2 bajty)

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.



6.7. Nastavení výstupu bzučáku při detekci karty

Tento příkaz nastaví výstup bzučáku při detekci karty. Výchozí výstup je zapnutý.

Formát příkazu (5 bajtů)

Příkaz	Třída	INS	P1	P2	Le
Nastavení výstupu bzučáku při detekci karty	FFh	00h	52h	PollBuzzStatus	00h

Kde:

P2 PollBuzzStatus

00h: Bzučák se při detekci karty NEzapne FFh:

Bzučák se při detekci karty zapne

Formát odpovědi (2 bajty)

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

7.0. Základní tok programu pro bezkontaktní aplikace

Krok 0. Spustíte aplikaci. Čtečka provede PICC Polling a bude průběžně vyhledávat značky.

Po nalezení a detekci značky se do počítače odešle odpovídající ATR. Musíte se ujistit, že byl nastaven příkaz PC/SC Escape. Další podrobnosti **naleznete v dodatku A**.

Krok 1. Nejprve je třeba připojit "ACR122U PICC Interface".

Krok 2. Získejte přístup k PICC odesláním příkazů APDU.

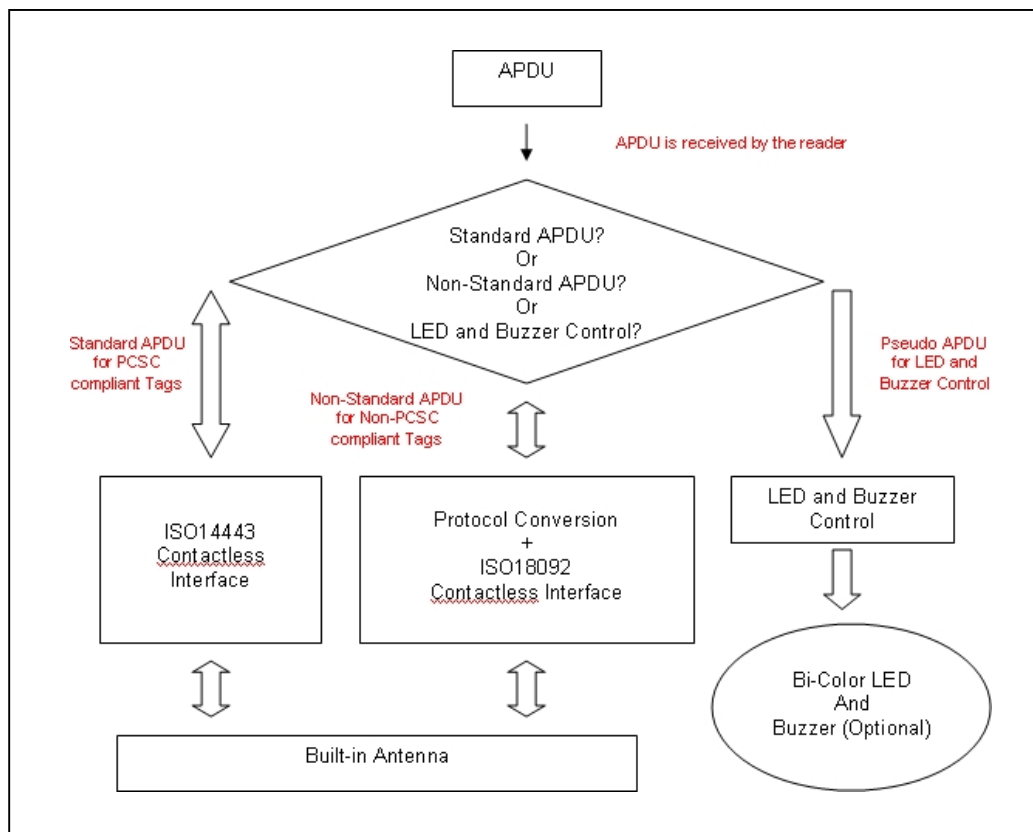
:

:

Krok N. Odpojte "ACR122U PICC Interface". Vypněte aplikaci.

Poznámky:

1. Pro úsporu energie lze anténu vypnout.
 - Vypněte napájení antény: FF 00 00 00 04 D4 32 01 00h
 - Zapněte napájení antény: FF 00 00 00 04 D4 32 01 01h
2. Zpracování standardních a nestandardních APDU.
 - PICC, které používají standardní APDU: ..., MIFARE ... atd.
 - PICC, které používají nestandardní APDU: FeliCa, Topaz .. atd.



Obrázek 3: Základní tok programu pro bezkontaktní aplikace



1. Pro rozhraní ACR122U PICC se používá protokol ISO 7816 T=1.
 - PC → Čtečka: Vydejte čtenáři APDU.
 - Čtečka → PC: Vrací se data odpovědi.

7.1. Jak přistupovat ke značkám kompatibilním s PC/SC (ISO 14443-4)?

V podstatě všechny karty kompatibilní s ISO 14443-4 (PICC) by rozuměly APDU ISO 7816-4. Čtečka ACR122U musí pouze komunikovat s kartami kompatibilními s ISO 14443-4 prostřednictvím výměny APDU a odpovědí ISO 7816-4. ACR122U bude interně zpracovávat protokoly ISO 14443 částí 1-4.

Tagy MIFARE 1K, 4K, Mini a Ultralight jsou podporovány prostřednictvím emulace T=CL. Stačí s tagy MIFARE zacházet jako se standardními tagy ISO 14443-4. Další informace naleznete v tématu: **PICC Commands for MIFARE Classic Memory Tags**.

Formát APDU ISO 7816-4

Příkaz	Třída	INS	P1	P2	Lc	Vstupní data	Le
ISO 7816 Část 4 Příkaz	-	-	-	-	Délka dat In	-	Očekávaná délka dat odpovědi

Formát odpovědi ISO 7816-4 (data + 2 bajty)

Reakce	Data Out		
Výsledek	Údaje o odezvě	SW1	SW2

Kódy odpovědi

Výsledky	SW1 SW2	Význam
Úspěch	90 00h	Operace byla úspěšně dokončena.
Chyba	63 00h	Operace se nezdařila.

Typická posloupnost může být následující:

1. Předložení štítku a připojení rozhraní PICC
2. Čtení/aktualizace paměti značky

1. Připojení štítku
2. Odeslat APDU, Získat výzvu.

<< 00 84 00 00 08h

>> 1A F7 F3 1B CD 2B A9 58h [90 00h]

Poznámka: Pro značky ISO14443-4 typu A lze ATS získat pomocí APDU "FF CA 00 00 01h".



7.2. Jak přistupovat k tagům MIFARE DESFire (ISO 14443-4)?

MIFARE® DESFire® podporuje režimy ISO 7816-4 APDU Wrapping a Native. Jakmile je MIFARE DESFire Tag aktivován, první APDU odeslaná do DESFire Tagu určí "Command Mode". Pokud je první APDU ve formátu "Native Mode", musí být i ostatní APDU ve formátu "Native Mode". Podobně pokud je první APDU "ISO 7816-4 APDU Wrapping Mode", musí být ostatní APDU ve formátu "ISO 7816-4 APDU Wrapping Mode".

Příklad 1: MIFARE DESFire ISO 7816-4 APDU Wrapping

Čtení 8 bajtů náhodného čísla z PICC ISO 14443-4 typu A (DESFire) APDU =

{90 0A 00 00 01 00 00h}

Class = 90; INS = 0A (instrukce DESFire); P1 = 00h; P2 = 00h Lc
= 01h; Data In = 00h; Le = 00h (Le = 00h pro maximální délku)

Odpověď: 7B 18 92 9D 9A 25 05 21h [\$91AFh]

Stavový kód [91 AFh] je definován ve specifikaci DESFire. Další podrobnosti naleznete ve specifikaci DESFire.

Příklad 2: Řetězení úrovní rámců MIFARE DESFire (režim zabalení podle ISO 7816)

V tomto příkladu musí aplikace provést "Frame Level Chaining". Pro získání verze karty MIFARE DESFire.

Krok 1: Odesláním APDU {90 60 00 00 00h} získáte první rámeček. INS=60

Odpověď: 04 01 01 00 02 18 05 91 AFh [\$91AFh]

Krok 2: Odesláním APDU {90 AF 00 00 00h} pro získání druhého rámečku. INS=AF

Odpověď: AFh [\$91AFh]: 04 01 01 00 06 18 05 91 AFh

Krok 3: Odesláním APDU {90 AF 00 00 00h} pro získání posledního rámečku.

INS=AFh Odpověď: 04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26 04 91 00h

[\$9100h]

Příklad 3: Nativní příkaz MIFARE DESFire

Pokud zjistíme, že nativní příkazy DESFire se lépe zpracovávají, můžeme čtečku poslat nativní příkazy DESFire bez obalu ISO 7816.

Čtení 8 bajtů náhodného čísla z PICC ISO 14443-4 typu A (DESFire) APDU =

{0A 00h}

Odpověď: D7h [\$1DD7h]

V něm je první bajt "AF" stavový kód vrácený kartou MIFARE DESFire. Data uvnitř blanketu [\$1DD7] může aplikace jednoduše ignorovat.



Příklad 4: MIFARE DESFire Frame Level Chaining (Nativní režim)

V tomto příkladu musí aplikace provést "Frame Level Chaining". Pro získání verze karty MIFARE DESFire.

Krok 1: Odeslání APDU {60h} pro získání prvního rámce.

INS=60h Odpověď: AF 04 01 01 00 02 18 05h[\$1805h]

Krok 2: Odeslání APDU {AFh} pro získání druhého rámce. INS=AFh

Odpověď: AF 04 01 01 00 06 18 05h[\$1805h]

Krok 3: Odeslání APDU {AFh} pro získání posledního rámce.

INS=AFh Odpověď: 00 04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26
04h[\$2604h]

Poznámka: V nativním režimu DESFire se stavový kód [90 00h] nepřidá k odpovědi, pokud je délka odpovědi větší než 1. Pokud je délka odpovědi menší než 2, přidá se stavový kód [90 00h], aby byl splněn požadavek PC/SC. Minimální délka odpovědi je 2.



7.3. Jak získat přístup ke značkám FeliCa (ISO 18092)?

Typická posloupnost může být následující:

1. Předložte štítek FeliCa a připojte rozhraní PICC.
2. Čtení/aktualizace paměti značky.

Krok 1) Připojte štítek.

ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 **F0 11** 00 00 00 00 8Ah

V němž,

F0 11 = FeliCa 212K

Krok 2) Čtení bloku paměti **bez použití pseudo APDU**.

<< 10 06h [8bajtové NFC ID] 01 09 01 01 80 00h

>> 1D 07h [8bajtové ID NFC] 00 00 01 00 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AAh [90 00h]

nebo

Krok 2) Načtení bloku paměti **pomocí pseudo APDU**.

<< **FF 00 00 00** [13] **D4 40 01** 10 06 [8-bajtové NFC ID] 01 09 01 01 80 00h

V němž,

[13] je délka pseudodat "**D4 40 01**.. 80 00h"

D4 40 01h je příkaz pro výměnu dat

>> **D5 41 00** 1D 07h [8bajtové ID NFC] 00 00 01 00 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AAh [90 00h]

V tomto případě je **D5 41 00h** odpověď na výměnu dat.

Poznámka: *ID NFC lze získat pomocí APDU "FF CA 00 00 00h".*

Podrobnější informace naleznete ve specifikaci FeliCa.

7.4. Jak přistupovat ke štítkům NFC fóra typu 1 (ISO 18092)?

Příkladem těchto značek jsou značky Jewel a Topaz.

Typická posloupnost může být následující:

1. Předložte značku Topaz a poté připojte rozhraní PICC.
2. Čtení/aktualizace paměti značky.

Krok 1) **Připojte štítek.**

ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 **F0 04** 00 00 00 00 9Fh

V němž **F0 04** = Topaz

Krok 2) **Čtení adresy paměti 08h (blok 1: Byte-0) bez použití pseudo APDU**

<< **01 08h**

>> **18h** [90 00h]

V tom, Response Data = **18h**

nebo

Krok 2) **Načtení adresy paměti 08h (blok 1: Byte-0) pomocí pseudo APDU**

<< **FF 00 00 00 [05] D4 40 01 01 08h**

V němž,

[05h] je délka pseudo APDU Data "**D4 40 01 01 08h**" **D4 40 01h**

je příkaz DataExchange.

01 08h jsou data, která mají být odeslána do tagu.

>> **D5 41 00 18h** [90 00h]

V tom, Response Data = **18h**

Tip: Chcete-li přečíst celý obsah paměti značky

<< **00h**

>> **11 48 18 26 .. 00h** [90 00h]

Krok 3) **Aktualizujte adresu paměti 08h (blok 1: Byte-0) pomocí dat FFh.**

<< **53 08 FFh**

>> **FFh** [90 00h]

V tom, Response Data = **FFh**


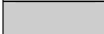

Adresa paměti = číslo bloku * 8 + číslo bajtu

Např. adresa paměti 08h (hex) = 1 x 8 + 0 = blok 1: Byte-0 = Data0

Např. adresa paměti 10h (hex) = 2 x 8 + 0 = blok 2: Byte-0 = Data8

HR0	HR1
11 _h	xx _h

Type	Block No.	Byte-0 (LSB)	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7 (MSB)	Lockable
UID	0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
Data	1	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Yes
Data	2	Data8	Data9	Data10	Data11	Data12	Data13	Data14	Data15	Yes
Data	3	Data16	Data17	Data18	Data19	Data20	Data21	Data22	Data23	Yes
Data	4	Data24	Data25	Data26	Data27	Data28	Data29	Data30	Data31	Yes
Data	5	Data32	Data33	Data34	Data35	Data36	Data37	Data38	Data39	Yes
Data	6	Data40	Data41	Data42	Data43	Data44	Data45	Data46	Data47	Yes
Data	7	Data48	Data49	Data50	Data51	Data52	Data53	Data54	Data55	Yes
Data	8	Data56	Data57	Data58	Data59	Data60	Data61	Data62	Data63	Yes
Data	9	Data64	Data65	Data66	Data67	Data68	Data69	Data70	Data71	Yes
Data	A	Data72	Data73	Data74	Data75	Data76	Data77	Data78	Data79	Yes
Data	B	Data80	Data81	Data82	Data83	Data84	Data85	Data86	Data87	Yes
Data	C	Data88	Data89	Data90	Data91	Data92	Data93	Data94	Data95	Yes
Reserved	D									
Lock/Reserved	E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	

	Reserved for internal use
	User Block Lock & Status
	OTP bits

Obrázek 4: Mapa paměti Topaz

Podrobnější informace naleznete ve specifikaci Jewel a Topaz.



7.5. Získání aktuálního nastavení bezkontaktního rozhraní

Krok 1. Příkaz Get Status.

<< FF 00 00 00 02 D4 04h

>> D5 05h [Err] [Field] [NbTg] [Tg] [BrRx] [BrTx] [Type] 80 90 00h

Nebo pokud v poli není žádná značka

>> D5 05 00 00 00 80 90 00h

[Err] je kód chyby odpovídající poslední zjištěné chybě.

Pole označuje, zda je přítomno a detekováno externí RF pole (pole = 01h), nebo ne (pole = 00h).

[NbTg] je počet cílů. Výchozí hodnota je 1.

[Tg]: logické číslo

[BrRx] : přenosová rychlost při

příjmu 00h: 106 Kbps

01h: 212 Kb/s

02h: 424 Kb/s

[BrTx] : přenosová rychlost při

přenosu 00h: 106 Kbps

01h: 212 Kb/s

02h: 424 Kb/s

[Type]: typ modulace 00h:

ISO 14443 nebo MIFARE

10h: FeliCa

01h: Aktivní režim

02h: Innovision Jewel tag

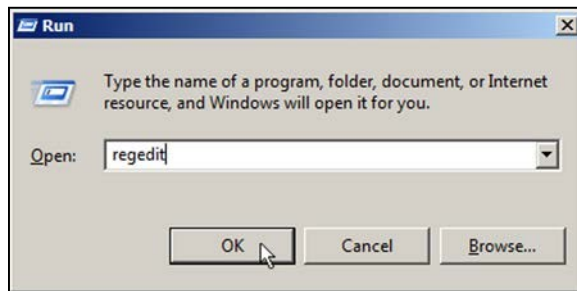
Dodatek A. ACR122U Příkaz PC/SC Escape

1. Vyberte "ACS ACR122U PICC Interface 0".
2. Zvolte "Shared Mode" (sdílený režim), pokud je rozhraní ACR122U PICC již připojeno, nebo "Direct Mode" (přímý režim), pokud rozhraní ACR122U PICC připojeno není.
3. Stisknutím tlačítka **Connect** navážete spojení mezi počítačem a čtečkou ACR122U.
4. Do textového pole příkazu zadejte "3500".
5. Zadejte PC/SC Escape Command, např. "FF 00 48 00 00h" a stiskněte tlačítko "Send" pro odeslání příkazu do čtečky. **#Zjištění verze firmwaru**
6. Stisknutím tlačítka **Odpojit** přerušíte spojení.
7. Chcete-li odeslat nebo přijmout **příkazy Escape** do čtečky, postupujte podle následujících pokynů.
8. IOCTL dodavatele pro příkaz **Escape** je definován takto:

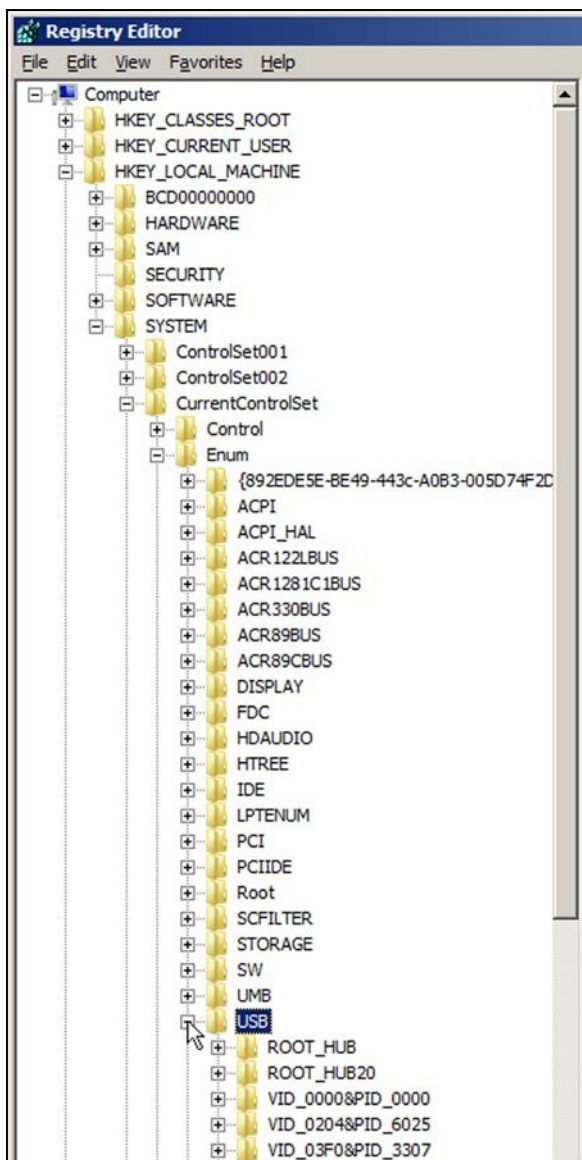
```
#define IOCTL_CCID_ESCAPE SCARD_CTL_CODE(3500)
```

Následující pokyny obsahují výčet kroků pro povolení příkazu PC/SC Escape:

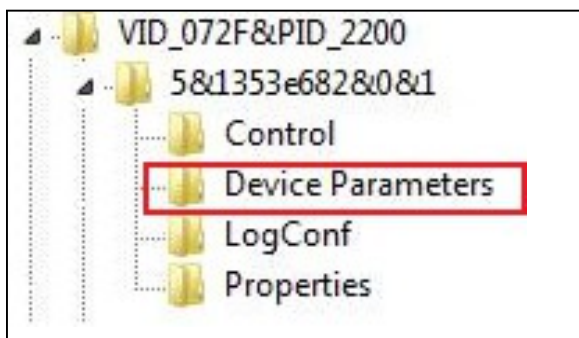
1. Spustíte příkaz "regedit" v nabídce "Spustit příkaz" systému Windows.



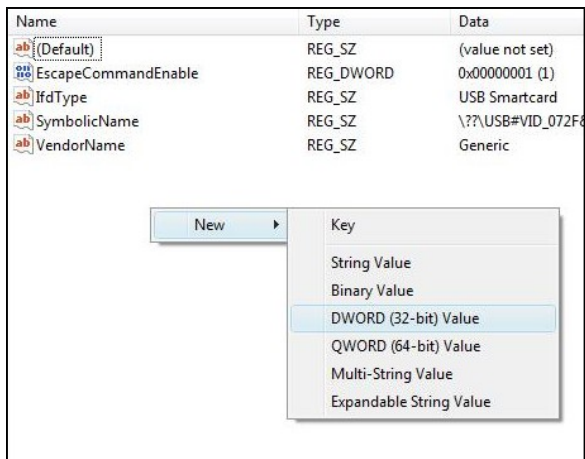
2. Přidejte DWORD "EscapeCommandEnable" pod HKLM\SYSTEM\CCS\Enum\USB\Vid_072F&Pid_90CC\Parametry zařízení
Pro systém Microsoft Vista je cesta následující:
Počítač\HKEY_LOCAL_MACHINE\SYSTEMS\CurrentControlSet\Enum\USB



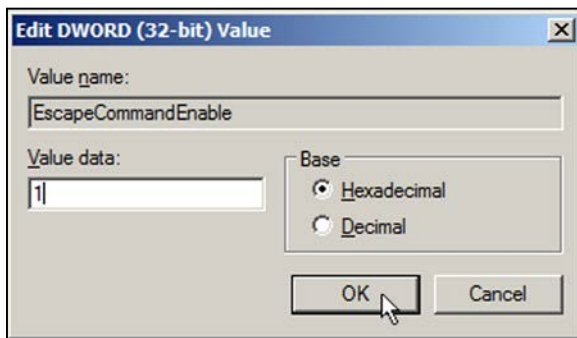
3. Hledejte: VID_072F&PID_2200 a pak tento uzel rozbalte. Podívejte se do části Parametry zařízení.



4. Vytvoření položky DWORD (32bitové) s názvem: EscapeCommandEnable



5. Chcete-li změnit hodnotu EscapeCommandEnable, dvakrát klikněte na položku a do údaje Value zadejte 1 se základem nastaveným v šestnáctkové soustavě.

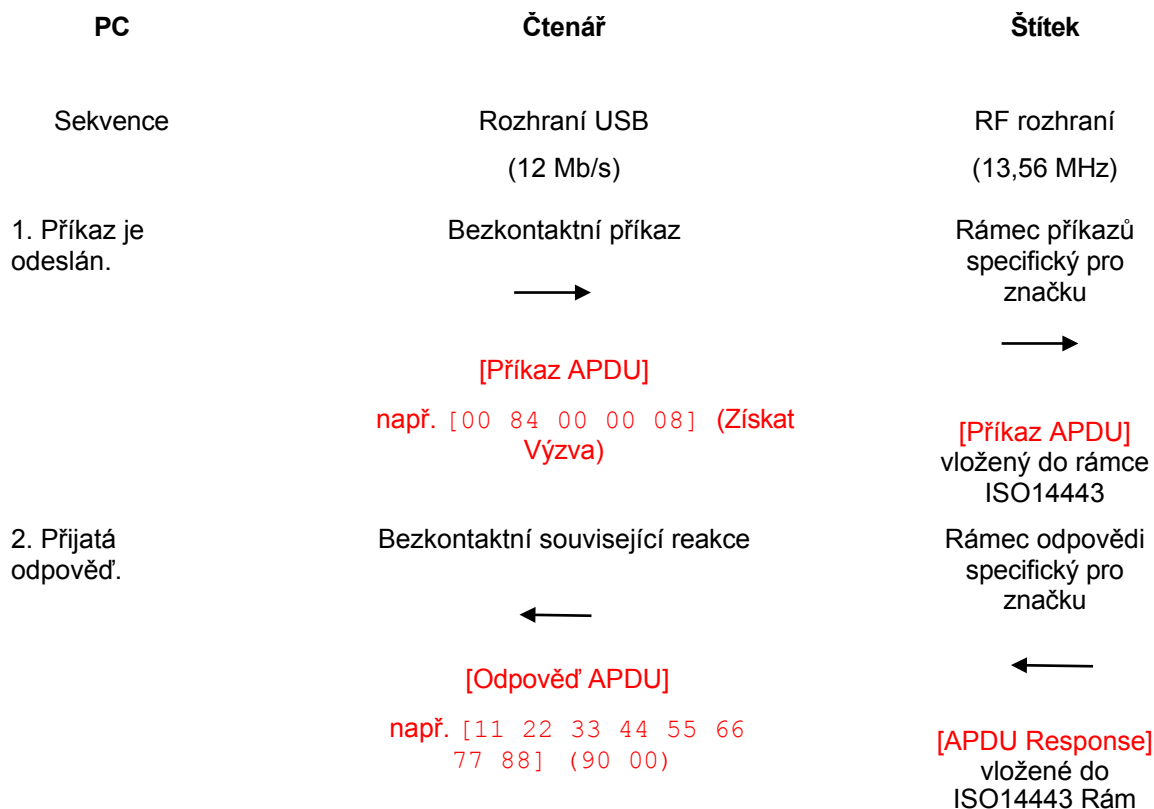




Dodatek B. Tok příkazů a odpovědí APDU pro štítky kompatibilní s ISO 14443

Předpokládáme, že se používá značka ISO 14443-4 typu B.

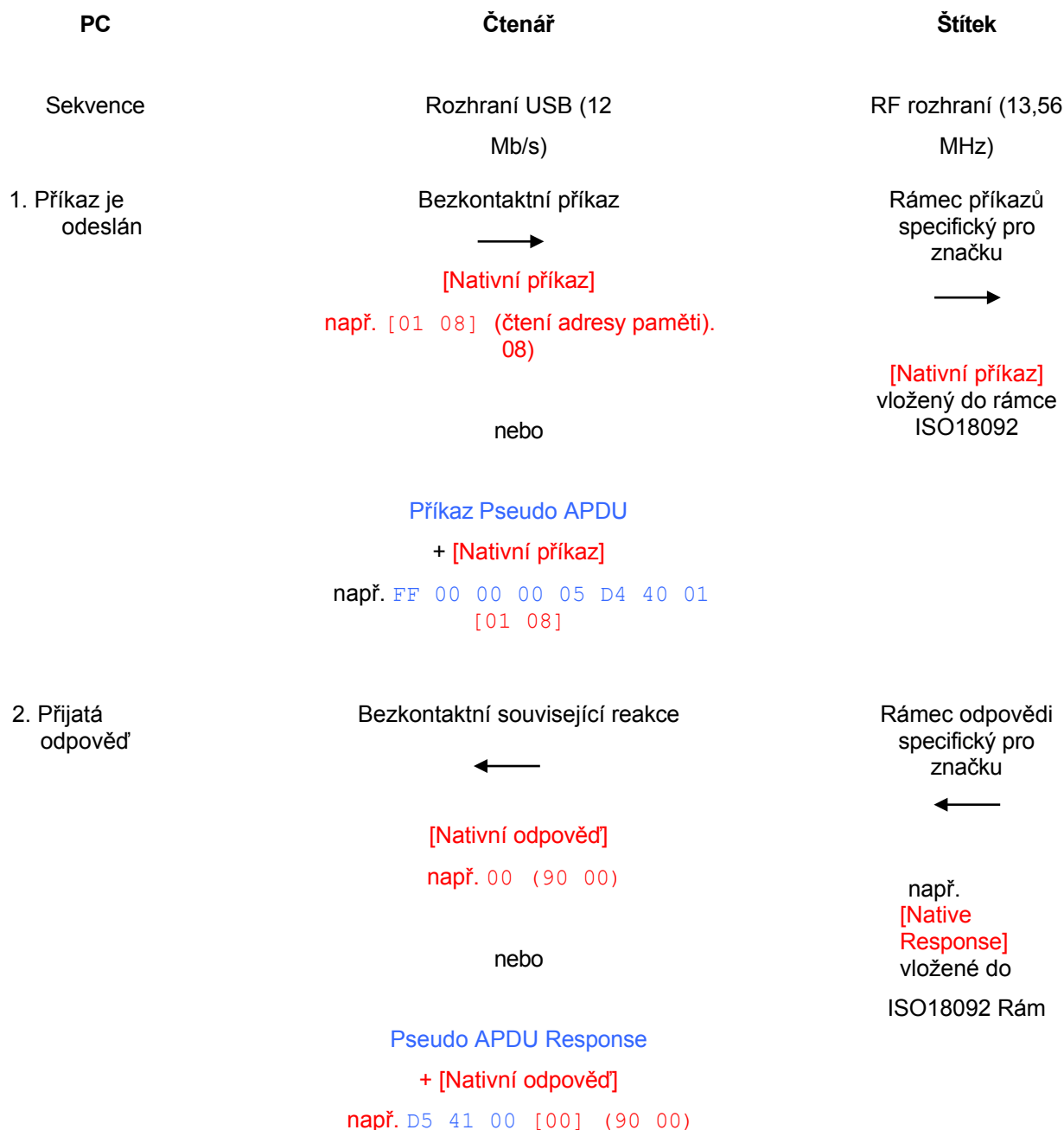
<< Typický tok příkazů a odpovědí APDU >>



Dodatek C. Tok příkazů APDUa odpovědí pro značky kompatibilní s ISO 18092

Poznámka: Předpokládáme, že se používá značka TOPAZ.

<< Typický tok příkazů a odpovědí APDU >>



Dodatek D. Chybové kódy

Kód chyby	Chyba
00h	Žádná chyba
01h	Time Out, cíl neodpověděl
02h	Bezkontaktní UART detekoval chybu CRC.
03h	Bezkontaktní UART detekoval chybu parity.
04h	Během operace MIFARE anti-collision/select byl zjištěn chybný počet bitů.
05h	Chyba rámování při operaci MIFARE
06h	Během bitové antikolize při rychlosti 106 Kb/s byla zjištěna abnormální bitová kolize.
07h	Nedostatečná velikost komunikační vyrovnávací paměti
08h	Bezkontaktní UART detekoval přetečení RF Bufferu (bit BufferOvfl registru CL_ERROR).
0Ah	V aktivním komunikačním režimu nebylo RF pole včas zapnuto protistranou (jak je definováno v normě NFCIP-1).
0Bh	Chyba protokolu RF (viz odkaz [4], popis registru CL_ERROR)
0Dh	Chyba teploty: vnitřní teplotní čidlo zjistilo přehřátí, a proto automaticky vypnulo ovladače antény.
0Eh	Přetečení vnitřní vyrovnávací paměti
10h	Neplatný parametr (rozsah, formát, ...)
12h	Protokol DEP: Čip nakonfigurovaný v cílovém režimu nepodporuje příkaz přijatý od iniciátora (přijatý příkaz není jedním z následujících: ATR_REQ, WUP_REQ, PSL_REQ, DEP_REQ, DSL_REQ, RLS_REQ, ref. [1]).
13h	Protokol DEP / MIFARE / ISO/IEC 14443-4: Formát dat neodpovídá specifikaci. V závislosti na použitém RF protokolu může být: <ul style="list-style-type: none"> • Špatná délka přijatého rámce RF, • Nesprávná hodnota PCB nebo PFB, • Neplatný nebo neočekávaný přijatý rámec RF, • NAD nebo DID inkoherence.
14h	MIFARE: Chyba autentizace
23h	ISO/IEC 14443-3: Kontrolní bajt UID je chybný
25h	Protokol DEP: Neplatný stav zařízení, systém je ve stavu, který neumožňuje operaci.
26h	Operace není v této konfiguraci povolena (rozhraní hostitelského řadiče)
27h	Tento příkaz není přijatelný vzhledem k aktuálnímu kontextu čipu (iniciátor vs. cíl, neznámé cílové číslo, cíl není v dobrém stavu, ...).
29h	Čip nakonfigurovaný jako cíl byl iniciátorem uvolněn.
2Ah	Pouze ISO/IEC 14443-3B: ID karty se neshoduje, což znamená, že očekávaná karta byla vyměněna za jinou.



Kód chyby	Chyba
2Bh	Pouze ISO/IEC 14443-3B: dříve aktivovaná karta zmizela.
2Ch	Nesoulad mezi iniciátorem NFCID3 a cílem NFCID3 v DEP 212/424 kbps passive.
2Dh	Byla detekována nadproudová událost.
2Eh	NAD chybí v rámečku DEP

Tabulka 7: Chybové kódy

Příloha E. Vzorové kódy pro nastavení LED diody

Příklad 1: Čtení stávajícího stavu LED.

// Předpokládejme, že červená i zelená LED dioda jsou zpočátku vypnuté //

// Není odkaz na bzučák //

APDU = "FF 00 40 00 04 00 00 00 00h"

Odpověď = "90 00h". Červená a zelená LED dioda jsou vypnuté.

Příklad 2: Rozsvícení červené a zelené barvy LED.

// Předpokládejme, že červená i zelená LED dioda jsou zpočátku vypnuté //

// Není odkaz na bzučák //

APDU = "FF 00 40 0F 04 00 00 00 00h"

Odpověď = "90 03h". Červená a zelená LED dioda svítí,

Chcete-li vypnout červenou i zelenou LED, APDU = "FF 00 40 0C 04 00 00 00 00h".

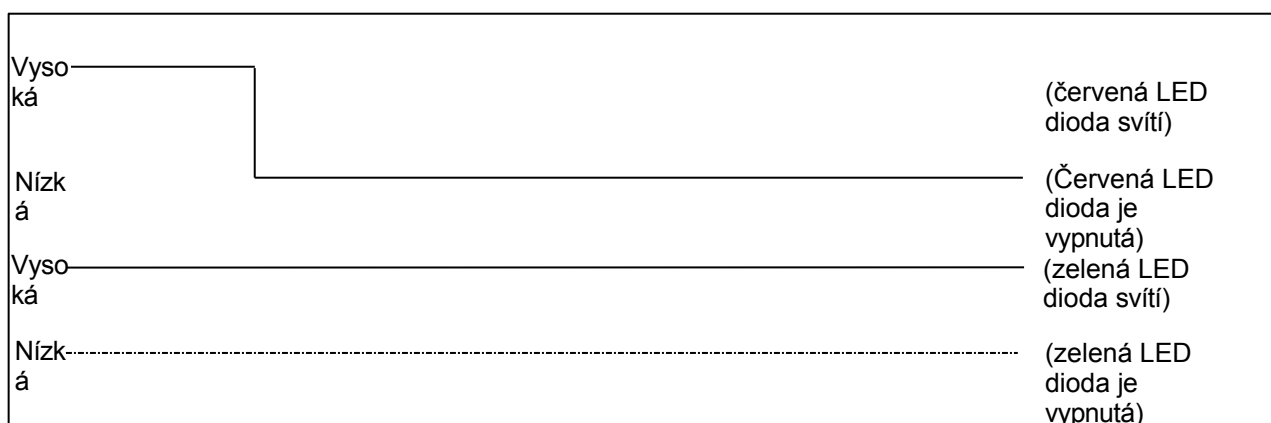
Příklad 3: Chcete-li vypnout pouze LED diodu červené barvy a ponechat LED diodu zelené barvy beze změny.

// Předpokládejme, že červená i zelená LED dioda jsou zpočátku zapnuté //

// Není odkaz na bzučák //

APDU = "FF 00 40 04 04 00 00 00 00h"

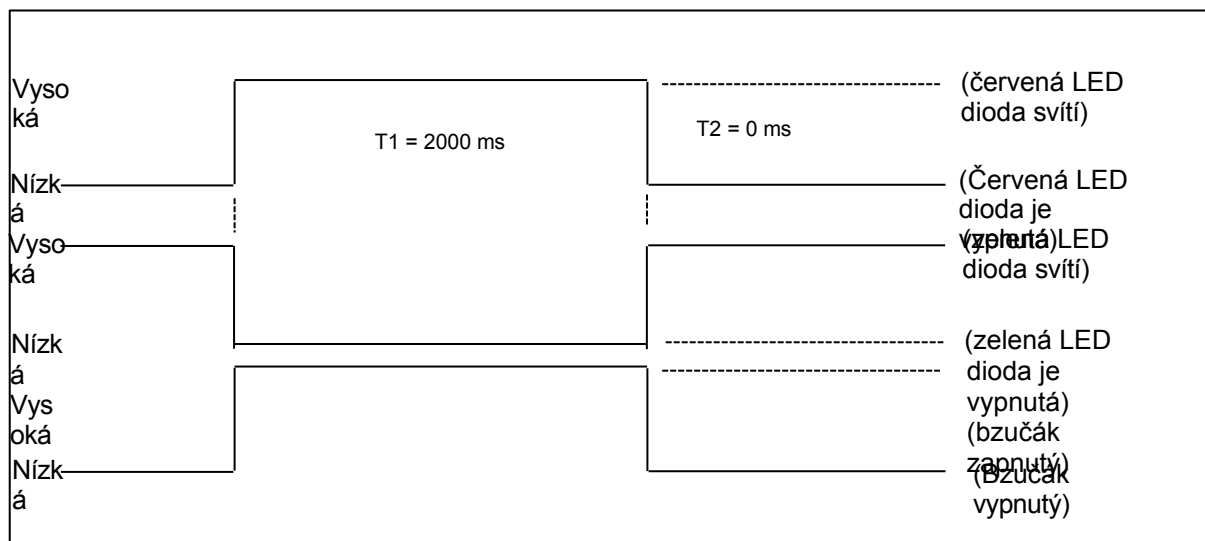
Odpověď = "90 02h". Zelená LED dioda se nemění (svítí); červená LED dioda je vypnutá,



Příklad 4: Rozsvícení červené kontrolky LED na 2 sekundy. Poté se vrátí do výchozího stavu.

// Předpokládejte, že červená LED dioda je zpočátku vypnutá, zatímco zelená LED dioda je zpočátku zapnutá. //

// Červená LED dioda a bzučák se během doby trvání T1 rozsvítí, zatímco zelená LED dioda během doby trvání T1 zhasne. //



1 Hz = 1000 ms Časový interval = 500 ms ON + 500 ms

OFF Doba trvání T1 = 2000 ms = 14h

T2 Doba trvání = 0 ms =

00h Počet opakování = 01h

Odkaz na bzučák = 01h

APDU = "FF 00 40 50 04 14 00 01 01h"

Odpověď = "90 02h"

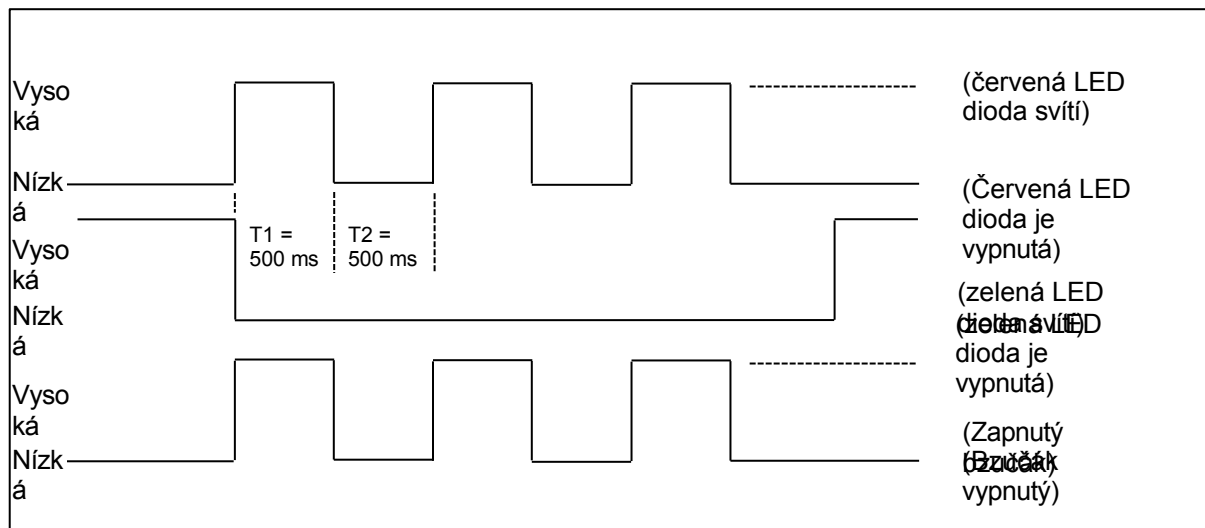
Příklad 5: Chcete-li, aby červená LED blikala třikrát frekvencí 1 Hz. Poté se vrátí do výchozího stavu.

// Předpokládejte, že červená LED dioda je zpočátku vypnutá, zatímco zelená LED dioda je zpočátku zapnutá. //

// Počáteční stav blikání červené LED je ON. Bliká pouze červená LED dioda.

// Bzučák se zapne během doby trvání T1, zatímco zelená LED dioda se vypne během doby trvání T1 i T2.

// Po blikání se rozsvítí zelená LED dioda. Červená LED dioda se po blikání vrátí do původního stavu //



1 Hz = 1000 ms Časový interval = 500 ms ON + 500 ms

OFF Doba trvání T1 = 500 ms = 05h

T2 Doba trvání = 500 ms =

05h Počet opakování = 03h

Odkaz na bzučák = 01h

APDU = "FF 00 40 50 04 05 05 03 01h"

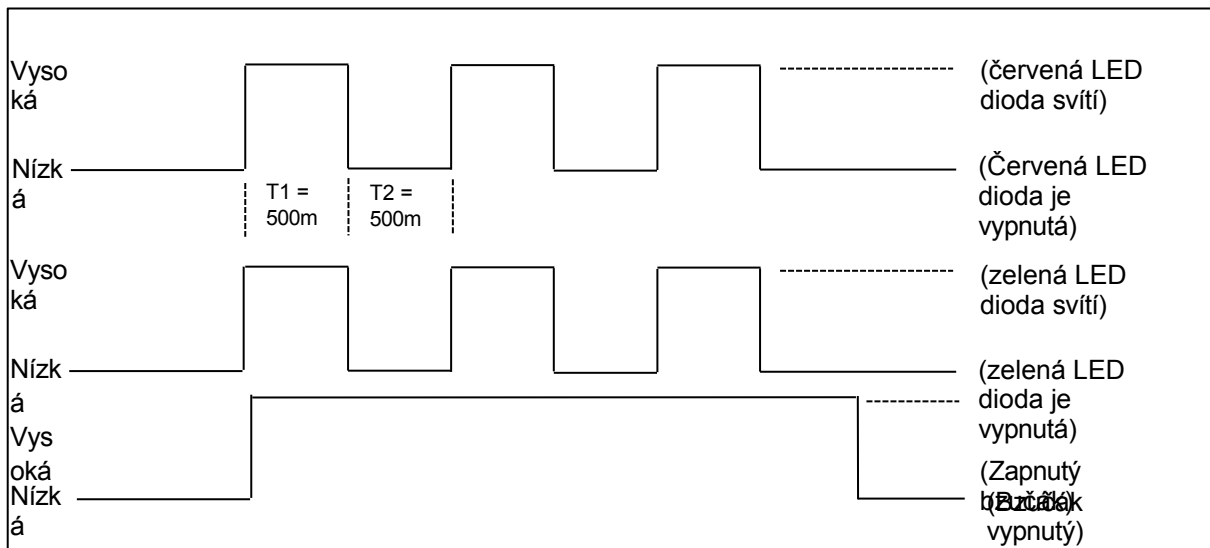
Odpověď = "90 02h"

Příklad 6: Chcete-li, aby červená a zelená LED dioda třikrát blikly frekvencí 1 Hz.

// Předpokládejte, že červená i zelená LED dioda jsou zpočátku vypnuté. //

// Oba počáteční stavy červeného a zeleného blikání jsou zapnuté //

// Bzučák se zapne během doby trvání T1 i T2//



1 Hz = 1000 ms Časový interval = 500 ms ON + 500 ms

OFF Doba trvání T1 = 500 ms = 05h

T2 Doba trvání = 500 ms =

05h Počet opakování = 03h

Odkaz na bzučák = 03h

APDU = "FF 00 40 F0 04 05 05 03 03h"

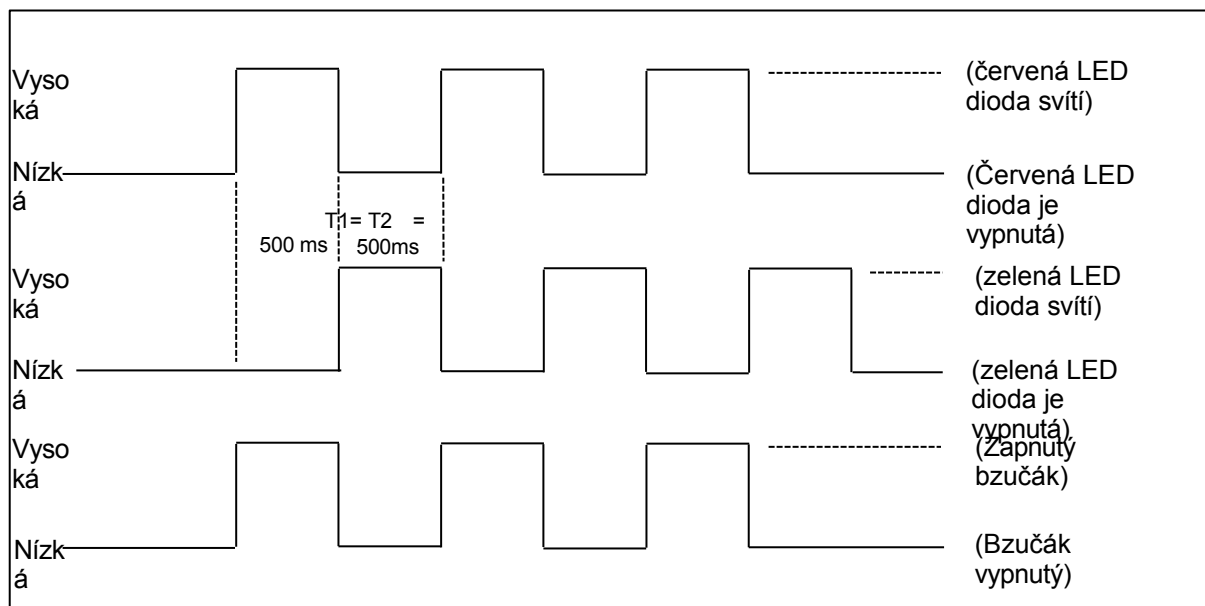
Odpověď = "90 00h"

Příklad 7: Chcete-li, aby červená a zelená LED dioda blikaly střídavě třikrát frekvencí 1 Hz.

// Předpokládejte, že červená i zelená LED dioda jsou zpočátku vypnuté. //

// Počáteční stav červeného blikání je ON; počáteční stav zeleného blikání je OFF //

// Bzučák se zapne po dobu trvání T1//



1 Hz = 1000 ms Časový interval = 500 ms ON + 500 ms

OFF Doba trvání T1 = 500 ms = 05h

T2 Doba trvání = 500 ms =

05h Počet opakování = 03h

Propojení s bzučákem = 01h

APDU = "FF 00 40 D0 04 05 05 03 01h"; odpověď = "90 00h".