

SW obsluha čteček (RD3, RD3B+, RD3M, RRD3x, ERD3x, RD4x, ARD4x)

Platné od verze sw: 1.154

V případě všech čteček xRD3x, je po instalaci USB driveru vytvořen virtuální COM port – viz samostatný návod pro připojení čtečky. Drivery naleznete na ALCS-line CD:\Driver\USB\CP21xx (nebo možno stáhnout na: <http://helpdesk.estelar.cz>). V případě TCP připojení čtečky je výchozí IP adresa 192.168.1.100, TCP port: 13000. Změna Komunikačních parametrů je možná přes web nebo telnet dle samostatného návodu.

Nastavení sériového portu:

Rychlost 9600 Bd, 8 bitů, 1 stop bit, parita žádná, řízení přenosu žádné
Propojkou JP1 lze nastavit rychlost na 19 200 Bd.

Funkce:

Čtečka vždy po přečtení karty automaticky odešle načtený kód na sériový port.

Data ze čtečky jsou v ASCII formátu:

16 znaků = kód karty + ukončovací znak <CR> (0x0D) + <LF> (0x0A)
pro některé variant čteček může být tento formát za provozu změněn
viz Nastavení formátu výstupních dat

Při změně stavu vstupu nebo relé posílá čtečka RRD3 automaticky informaci o této změně ve stejném formátu jako při vyžádaném dotazu – viz dále.

Pro ověření funkčnosti nebo přítomnosti čtečky je možno zapisovat dotazy:

Přes program Hercules SETUP utility je třeba znaky # a \$ zdvojit.

Znak '?' – čtečka znovu pošle poslední načtený kód karty

Znak '#' – *čtečka pošle svou identifikaci podle typu

např: "RRD3-xxx", kde xxx je číslo verze

Znak 'I' – ** čtečka pošle stav vstupů "I1x"<CR><LF>"I2x"

kde x = 0 pro neaktivní, nebo x = 1 pokud je aktivní (spojeno GND)

Znak 'R' – ** čtečka pošle stav relé "RF", nebo "RO" pokud je sepnuto

Každá odpověď od čtečky je ukončena znaky <CR> (0x0D) + <LF> (0x0A)

Jakýkoliv jiný znak zapsaný do čtečky bude vrácen nezměněn.

Interní funkce lze ovládat zapsáním příkazů:

Příkaz: '\$O<CR><LF>' – ** sepnutí relé

Příkaz: '\$F<CR><LF>' – * vypnutí relé

Příkaz: '\$A050<CR><LF>' – ** sepnutí relé na požadovaný čas 5s (max 99 sec)

kde xxx = doba sepnutí x100 ms

Příkaz: '\$G020<CR><LF>' – *počet bliknutí zelenou LED x 500ms (max.999)

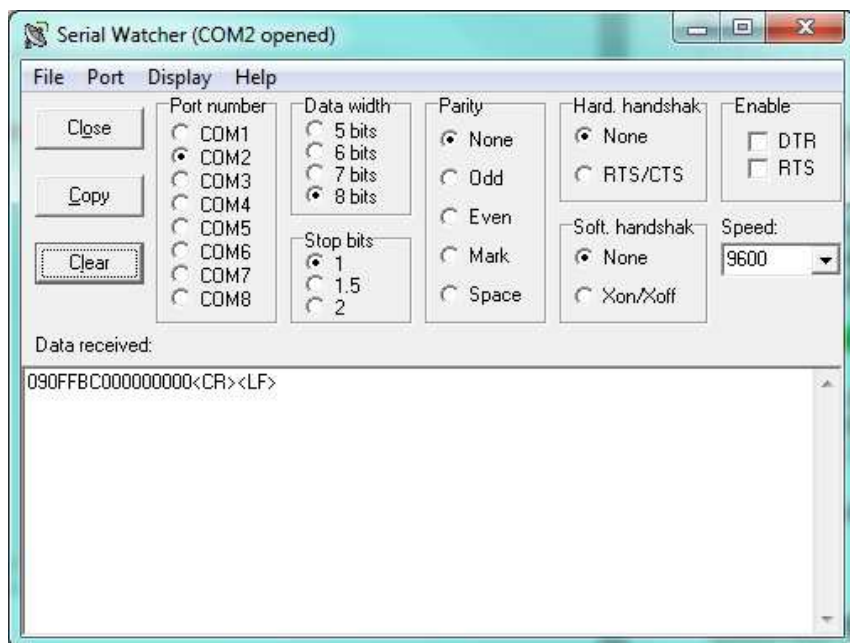
příklad: 20 krát. blik = 10sekund (přečtením karty se blikání ukončí)

Každý zápis příkazu musí být ukončen sekvencí: <CR> (0x0D) + <LF> (0x0A)

*) tyto příkazy jsou platné pouze pro čtečky RD3B+, RD3M, ERD3x, RRD3x od verze 1.139

***) tyto příkazy jsou platné pouze pro čtečky RRD3

Funkci čtečky lze otestovat například pomocí programu Hyperterminál, nebo podobných nástrojů běžně dostupných. Z webu <http://www.acsline.cz/cs/pro-partnery> lze stáhnout prográmk SerialWatcher (naleznete také na ACS-line CD:\SERVIS).



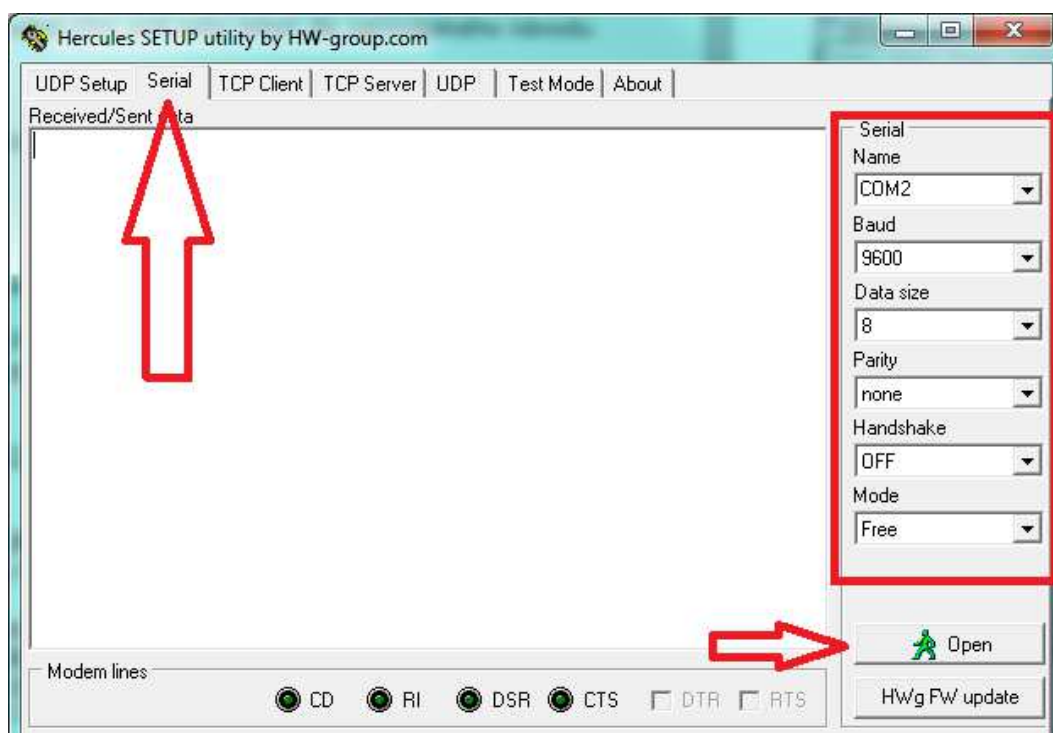
V terminálovém okně se po správné konfiguraci budou zobrazovat všechny přijaté znaky ze čtečky. V případě přiložení karty se vypíše kód v aktuálně nastavené interpretaci. V případě změny stavu vstupu například „I1“.

Pro zápis konfiguračních příkazů

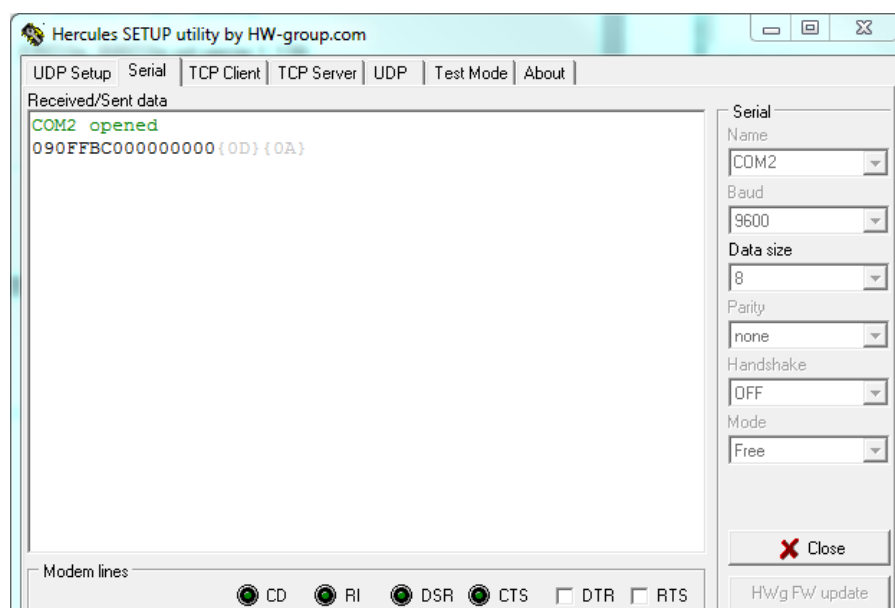
doporučujeme použít např. Hercules SETUP utility (www.HW-group.com).

Link pro stažení a podrobnosti k nastavení naleznete zde:

http://www.hw-group.com/products/hercules/index_cz.html



Po přiložení karty se vypíše kód karty ve výchozím nastavení.
Např. 090FFBC000000000



Odesláním příkazů vložených do políček Send proběhne změna konfigurace.

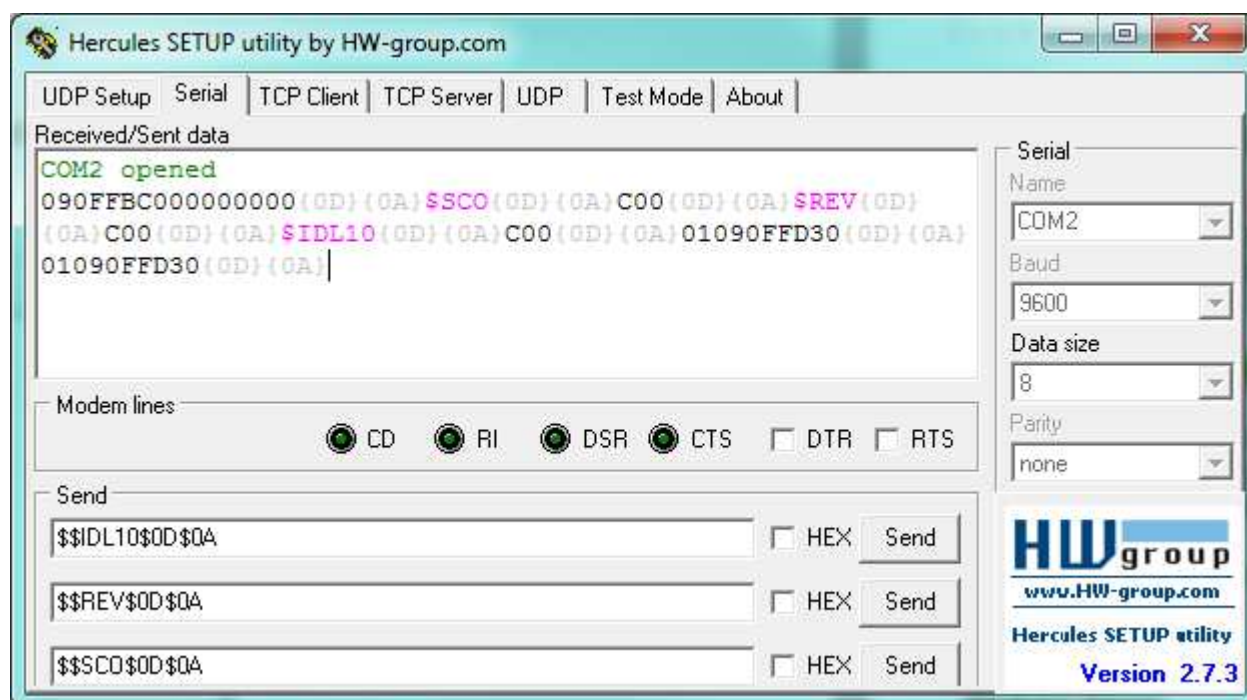
První znak \$ musí být zdvojen (\$\$), pokud je odeslán v textovém režimu. Naopak použití jednoho znaku \$ umožní zadání znaku jeho HEX hodnotou.

\$\$SCO\$0D\$0A - načítání z karty včetně site code (celkem 5 bytů)

\$\$REV\$0D\$0A - otočené pořadí bitů

\$\$IDL10\$0D\$0A - délka výstupního řetězce na 10 znaků

Výstup po změně konfigurace: 01090FFD30



Nastavení formátu výstupních dat a konfigurační parametry

Zapsáním konfigurační sekvence znaků lze upravit konverzi a výstupní formát načteného kódu ID média. Příkazy lze kombinovat (mimo ty které se logicky vylučují) pro dosažení požadovaného výsledku. Čtečka si nastavení bude pamatovat i po odpojení napájení (USB). Případný návrat do výrobního stavu lze uskutečnit příkazem DEF.

Každý zápis příkazu musí být následován sekvencí: <CR> (0x0D) + <LF> (0x0A)

Odpověď (potvrzení) po zápisu: C**ss**<CR> (0x0D) + <LF> (0x0A)

Odpověď při čtení: T**ss**<CR> (0x0D) + <LF> (0x0A)

Kde ss = návratový status s významem podle jednotlivých instrukcí (ss=00=OK)

Přijetí a zápis konfiguračního příkazu je indikováno krátkým svitem žluté LED a písknutím.

Výchozí původní nastavení je u parametrů označeno *).

Globální řízení:

'\$DEF<CR><LF>' – obnoví výrobní nastavení

'\$RST<CR><LF>' – provede restart zařízení – např. po změně konfigurace

'\$FST<CR><LF>' – odešle informaci jestli je karta přítomna v poli (T01= ANO, T00=NE)

'\$TMOxx<CR><LF>' – nastaví timeout pro přiložení karty (rozsah 01-99s, pro práci s bloky MIFARE)

Délka zpracovaných dat:

'\$SCO<CR><LF>' – kód karty obsahuje celkem 5 bytů z karty (EMmarin), nebo 7-bajtů (MIFARE)

'\$SCF<CR><LF>' – * kód karty obsahuje celkem 4 byty z karty (Emmarin i MIFARE)

'\$IDL05<CR><LF>' – kód karty se posílá jako 5 znaků (platí pouze pro DEC vyjádření)

'\$IDL08<CR><LF>' – kód karty se posílá jako 8 znaků – ořezáno/doplněno zprava

'\$IDL10<CR><LF>' – kód karty se posílá jako 10 znaků – ořezáno/doplněno zprava

'\$IDL12<CR><LF>' – kód karty se posílá jako 12 znaků – ořezáno/doplněno zprava

'\$IDL14<CR><LF>' – kód karty se posílá jako 14 znaků – ořezáno/doplněno zprava

'\$IDL16<CR><LF>' – * kód karty se posílá jako 16 znaků – ořezáno/doplněno zprava

Formát výstupních dat:

'\$W26<CR><LF>' – kód z karty prezentován jako ze čtečky s výstupem Wiegand 26

'\$W32<CR><LF>' – kód z karty prezentován jako ze čtečky s výstupem Wiegand 32

'\$W42<CR><LF>' – kód z karty prezentován jako ze čtečky s výstupem Wiegand 42

'\$DEC<CR><LF>' – kód z karty je převeden na dekadické vyjádření (vypne funkci HEX/DAL)

'\$HEX<CR><LF>' – * kód z karty je ve standartním HEX vyjádření (vypne funkci DEC/DAL)

'\$DAL<CR><LF>' – kód z karty je převeden na formát používaný u DALLAS čipů (vypne funkce HEX/DEC)

Pořadí bajtů/bitů:

'\$RBO<CR><LF>' – kód z karty je zpracován v opačném pořadí bytů

(byte načtený z karty jako poslední bude na první pozici)

'\$REV<CR><LF>' – kód z karty je zpracován v opačném pořadí bitů v horní a dolní

polovině bajtu (opakované odeslání příkazu zapne/vypne funkci, lze kombinovat)

Komunikace:

'\$CCOxx<CR><LF>' – čtečka vkládá do výstupního paketu řídicí znaky STX + ETX

'\$LOPxx<CR><LF>' – čtečka posílá opakovaně kód karty která je v poli. Indikováno svitem zelené LED.

'\$LOFxx<CR><LF>' – čtečka posílá opakovaně kód poslední načtené karty (i když není v poli)
(interval v rozsahu 01-99s, 00=vypnuto). Indikováno bliknutím zelené LED.

'\$LOTxx<CR><LF>' – počet odeslání paketů dle nastavení '\$LOF'
(rozsah 01-99x, 0=vypnuto, default=nekonečno, pokud je nastaven LOF)

Pro LOP, LOF může být aktivní vždy pouze jeden z parametrů.

Signalizace:

'\$BZO<CR><LF>' – *aktivace zvuku písače při načtení karty (výchozí nastavení)

'\$BZF<CR><LF>' – deaktivace zvuku písače při načtení karty

'\$BZLxx<CR><LF>' – nastavení délky zvuku písače při načtení karty (rozsah 01-99 x100ms)

'\$BZSxx<CR><LF>' – nastavení délky zvuku písače při zapnutí čtečky (rozsah 01-99s, 00=neaktivní)

'\$BAO<CR><LF>' – aktivace zvuku písače při požadavku na přiložení karty (pouze MIFARE)
nebo při zapsání příkazu pro blikání zelenou LED (\$Gxxx)

'\$BAF<CR><LF>' – *deaktivace této funkce (výchozí nastavení)

'\$CLTxxx<CR><LF>' – nastavení délky svitu LED po načtení karty (rozsah 001-255s)

Zákaznické nastavení (interní použití, nedokumentováno):

'\$USRXX<CR><LF>' – číslo zákaznické modifikace

- USR01 - autočtečka, změna indikace LED
- USR02 - na poslední pozici v dek. formátu se vkládá druhý bajt z ID v hexa
- USR03 – změna formátu TYCO
-

'\$RIDxx<CR><LF>' – číslo čtečky (rozsah adres xx=01-99, výstupní formát dat xx0000ccccccc)

Příkazy pro práci s kartami MIFARE (RD3M, RRD3M, ERD3M)

Legenda:

- [cc..] = kód karty ve formátu jak je odeslán při běžném čtení
- [bb] = číslo sektoru pro čtení/zápis
- [A/B] = typ klíče, KEY-A, nebo KEY-B (např. [A] nebo [B])
- [kk..] = klíč pro čtení nebo zápis – 12 znaků (např. A0A1A2A3A4A5)
- [id] = index interního klíče z eeprom 01-24, viz. Seznam default interních klíčů.
pokud je předáván klíč v poli [kk..] tak [id]=[00]
- [xx..] = datový blok - 16 bytů v HEX formátu
- [vv..] = hodnota value (4-bajtová hodnota v hex – např. 0000000A)
- [aa] = adresa pro zápis/čtení uživatelské oblasti EEPROM (viz. seznam default klíčů)
- [crc] = 4 znaky kontrolní součet dle popisu dále
- [ss] = status provedení operace (0 = OK, >0 = kód chyby)

Každý zápis příkazu musí být ukončen sekvencí: <CR> (0x0D) + <LF> (0x0A)
Čtečka vždy vrátí status zápisu, nebo načtená data v případě čtení

Výpočet CRC

Pro výpočet CRC součtu je použit algoritmus CRC-16-CCITT (X-modem).
polynom 0x1021 – (formát little endian -> DCBA) . Pro výpočet vhodné použít referenční kalkulátor na
<http://www.lammertbies.nl/comm/info/crc-calculation.html>

Organizace karty MIFARE

Paměťový prostor karty MIFARE 1kB je rozdělen na 16 sektorů x 4 bloky. Každý blok obsahuje 16 bajtů.
Jednotlivé bloky jsou adresovány v HEX rozsahu 00-3F.
Přístup do jednotlivých bloků karty je definován v tzv. sector traileru bloku.

Zápis/čteční sector traileru

'\$RDW[cc..][bb][A/B][id][xx..][crc]<CR><LF>' - zápis bloku, autorizace interním default klíčem
'\$RDW[cc..][bb][A/B][kk..][xx..][crc]<CR><LF>' - zápis bloku, autorizace předaným klíčem

Příklad použití:

'\$RDW0E88A3B833A01FFFFFFFFFFFF08778F00E0E1E2E3E4E57EDA<CR><LF>'

- UID 0E88A3B8, inicializace sector traileru – blok 33h, všechny 3 bloky jako VALUE (08778F)
nový klíč A= FFFFFFFFFF, nový klíč B =E0E1E2E3E4E5,
přístup v transport konfiguraci klíčem A (FFFFFFFF, id=01).

'\$RDW0E88A3B833B00E0E1E2E3E4E5A0A1A2A3A4A508778F00E0E1E2E3E4E5C971<CR><LF>'

- Změna přístupových klíčů v sector traileru - blok 33h
Nový klíč A = A0A1A2A3A4A5, nový klíč B=E0E1E2E3E4E5, přístup klíčem B=E0E1E2E3E4E5

POZOR NA ZÁPIS NESPRÁVNÉ HODNOTY DO SECTOR TRAILERU !
ZPŮSOBÍ TO NEVRATNÉ UZAMČENÍ CELÉHO SEKTORU.

Zápis/čteční DATA bloku

'\$RDR [cc..][bb][A/B][00][kk..][crc]<CR><LF>' - čtení bloku, autorizace klíčem [kk..], typ A/B
'\$RDR [cc..][bb][A/B][id][crc]<CR><LF>' - čtení interním klíčem [id], typ A/B

'\$RDW [cc..][bb][A/B][00][kk..][xx..][crc]<CR><LF>' - zápis bloku, autorizace klíčem [kk..], typ A/B
'\$RDW [cc..][bb][A/B][id][xx..][crc]<CR><LF>' - zápis bloku interním klíčem [id], typ A/B

Odpověď na čtení: T[bb][ss][xx..]<CR>+<LF>

Odpověď po zápisu: P[bb][ss]<CR>+<LF>

Příklad použití:

'\$RDR0E88A3B833A019135<CR><LF>' - čtení DATA bloku 33h klíčem A č. 1

'\$RDR0E88A3B833B00E0E1E2E3E4E5DE38<CR><LF>' - čtení DATA bloku 33h, klíč B E0E1E2E3E4E5

'\$RDW0E88A3B805B01KONTOLNI_BLOK_051FBC<CR><LF>'
- zápis DATA bloku 05h, klic B c.01, data "KONTOLNI_BLOK_05"

'\$RDW0E88A3B805A00FFFFFFFFFFFFFFFF000102030405060708090a0b0c0d0e0f10b875<CR><LF>'
- zápis DATA bloku 05h, klíčem A FFFFFFFFFF, data „000102030405060708090a0b0c0d0e0f10“

Operace s VALUE blokem – zápis / čtení / inkrement /dekrement

'\$RVR [cc..][bb][A/B][00][kk..][crc]<CR><LF>' - čtení VALUE, autorizace poskytnutým klíčem A/B
'\$RVR [cc..][bb][A/B][id][crc]<CR><LF>' - čtení VALUE, interním klíčem [id]

'\$RVW [cc..][bb][A/B][00][kk..][vv..][crc]<CR><LF>' - zápis VALUE, autorizace klíčem A/B
'\$RVW [cc..][bb][A/B][id][vv..][crc]<CR><LF>' - zápis VALUE, interním klíčem [id]

'\$RVI [cc..][bb][A/B][id][vv..][crc]<CR><LF>' - inkrement interním klíčem, [vv..] = inkrement
'\$RVI [cc..][bb][A/B][00][kk..][vv..][crc]<CR><LF>' - inkrement poskytnutým klíčem, [vv..]=inkrement

'\$RVD [cc..][bb][A/B][id][vv..][crc]<CR><LF>' - dekrementace interním klíčem, [vv..] = dekrement
'\$RVD [cc..][bb][A/B][00][kk..][vv..][crc]<CR><LF>' - inkrement poskytnutým klíčem, [vv..]=inkrement

Odpověď po zápisu/ inkrement/dekrement: P[bb][ss]<CR>+<LF>

Odpověď po čtení: T[bb][ss][vv..]<CR>+<LF>

Příklad použití:

\$RVR0E88A3B816B0B7D04<CR><LF>
- čtení VALUE bloku 16h – interním klíčem B č. 11

\$RVR0E88A3B816B00D0D1D2D3D4D59041<CR><LF>
- čtení VALUE bloku 16h - klíčem B D0D1D2D3D4D5

\$RVW0E88A3B816A020000001BB54B<CR><LF>
- zápis do VALUE bloku - blok 16h, value = 0x0000001B, interním klíčem A č.02

\$RVW0E88A3B816B00D0D1D2D3D4D5000000AADA2C<CR><LF>

- zápis do VALUE bloku - blok 16h, value = 0x000000AA, klíčem B - D0D1D2D3D4D5

\$RVIOE88A3B816B0B0000000184BB<CR><LF>

- inkrement bloku 16h (+1), interní klíč B č.11 (0Bh)

\$RVIOE88A3B816A00D0D1D2D3D4D500000001BAA6<CR><LF>

- inkrement bloku 16 (+1), klíčem A D0D1D2D3D4D5

\$RVD0E88A3B816B0B000000011081<CR><LF>

- dekrement bloku 16h (-1), interní klíč B č.11 (0Bh)

\$RVD0E88A3B816A00D0D1D2D3D4D500000001D7F9<CR><LF>

- dekrement bloku 16 (-1), klíčem A D0D1D2D3D4D5

Výpis obsahu celé karty – DUMP

‘\$RWD<CR><LF>

Čtečka se pokusí autentizovat přístup do bloků default klíči uloženými v paměti EEPROM (pevné pozice 1-10 plus uživatelské pozice 11-24). Viz. seznam přístupových default klíčů.

Čtečka odpoví formátovaným výpisem celého paměťového prostoru karty.

Význam identifikátorů ATQA a SAK je v následující tabulce.

Card UID: 0E88A3B8
ACS Line: B8A3880E00000000
PICC type: MIFARE 1k ATQ: 0x0400 SAK: 0x08

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits	Key_A	Key_B
0F	3F	00	00	00	00	00	00	08	77	8F	00	00	00	00	00	00	00	[0 1 1]	01:FFFFFFFFFFFF	0B:PROTECTED
	3E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
	3D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
	3C	* 78	56	34	12	87	A9	CB	ED	78	56	34	12	3C	C3	3C	C3	[1 1 0]	[Value] 0x12345678	
0E	3B	00	00	00	00	00	00	08	77	8F	00	00	00	00	00	00	00	[0 1 1]	01:FFFFFFFFFFFF	0A:000000000000
	3A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
	39	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
	38	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
0D	37	PICC authent. error																		
0C	33	00	00	00	00	00	00	08	77	8F	00	00	00	00	00	00	00	[0 1 1]	02:A0A1A2A3A4A5	0C:PROTECTED
	32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
	31	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[1 1 0]	[Value]	
0B	2F	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]	01:FFFFFFFFFFFF	01:FFFFFFFFFFFF
	2E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]		
	2D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]		
	2C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]		

Legenda: číslo před klíčem A/B znamená pozici klíče v EEPROM (např. 02:A0A1A2A3A4A5)
hodnotu uživatelského klíče v EEPROM není možné zobrazit (např. 0C:PROTECTED)
[VALUE] = value block, pokud neuvedeno jedná se o data block
[Locked] = klíč nemá oprávnění ke čtení data bloku

Tabulka č.1: Návrátové kódy karet MIFARE ATQA + SAK

Typ karty MIFARE	ATQA	SAK	UID SIZE
MIFARE Ultralight CL1	0x4400	0x04	
MIFARE Ultralight CL2	0x4400	0x00	
MIFARE 1k	0x0400	0x08	4-Byte
MIFARE 4k	0x0200	0x18	4-Byte
MIFARE DESFire CL1	0x4403	0x24	7-Byte
MIFARE DESFire CL2	0x4403	0x20	7-Byte
MIFARE Plus 2k SL1/2/3	0x0400/0200	0x08/0x10/0x20	4-Byte
MIFARE Plus 4k SL1/2/3	0x0400/0200	0x18/0x11/0x20	4-Byte
MIFARE Plus 2k CL2 SL1/2/3	0x4400/4200	0x08/0x10/0x20	7-Byte
MIFARE Plus 4k CL2 SL1/2/3	0x4400/4200	0x18/0x11/0x20	7-Byte
JAVA KARTY			
JCOP31	0x0403	0x28	
JCOP31 v2.4.1	0x4800	0x20	
JCOP41 v2.2	0x4800	0x20	
JCOP41 v2.3.1	0x0400	0x28	

Správa přístupových klíčů

Defaultní přístupové klíče použité k autentizaci přístupu ke kartě jsou uloženy v interní paměti EEPROM čtecího obvodu MIFARE. Klíče lze do této paměti pouze zapisovat. Vyčtení klíče není možné.

Seznam default klíčů lze rozšířit zapsáním uživatelských klíčů, a tyto využívat pro zkrácenou autentizaci (bez poskytování klíče paketem).

Klíče uložené na pozicích 01-10 není možné změnit, ani na danou pozici zapsat.

Pro uložení uživatelských klíčů jsou určeny pozice 11-24.

Zápis klíčů do interní paměti EEPROM čtecího obvodu

'\$LKE[id][kk.][[crc]<CR><LF>' - zápis klíče do seznamu interních klíčů
'\$LKE[FF][kk.][[crc]<CR><LF>' - inicializace prostoru uživatelských klíčů - default

Odpověď po zápisu/čtení: P[id][ss]<CR>+<LF>

Čtení není možné.

Příklad použití:

\$LKEFFFFFFFFFFFFFFFF1925<CR><LF> - inicializace prostoru 11-24 default klíčem
\$LKE0BD0D1D2D3D4D5E318<CR><LF> - zápis klíče na pozici 11 (0bh) D0D1D2D3D4D5
\$LKE0CE0E1E2E3E4E58DBD<CR><LF> - zápis klíče na pozici 12 (0Ch) E0E1E2E3E4E5

Seznam default klíčů – interní EEPROM

Pozice klíče	klíč	poznámka
1	FFFFFFFFFFFF	default klíč A/B pro karty NXP
2	A0A1A2A3A4A5	default klíč A pro karty INFINEON
3	B0B1B2B3B4B5	default klíč B pro karty INFINEON
4	A0B0C0D0E0F0	
5	A1B1C1D1E1F1	
6	4D3A99C351DD	
7	1A982C7E459A,	
8	D3F7D3F7D3F7	
9	AABBCCDDEEFF	
10	000000000000	
11..24	default FFFFFFFFFF	pozice pro uživatelské klíče

Zápis/čtení uživatelských dat do paměti EEPROM čtecího obvodu

Pro uživatelské data je vymezen adresní prostor bloků 3-7.

Délka zapisovaných dat je vždy 16 bajtů.

Adresa bloku je pak 0x30, 0x40, 0x50, 0x60, 0x70

Zápis mimo vyhrazený adresní prostor není možný – tj. neprovede se.

'\$REW[aa][xx.][[crc]<CR><LF>' - zápis do bloku EEPROM, 16 bajtů
'\$RER[bb]<CR><LF>' - čtení bloku EEPROM, 16 bajtů

Odpověď po zápisu: P[bb][ss]<CR>+<LF>

Odpověď na čtení: T[bb][ss][xx.]<CR>+<LF>

Příklad použití:

\$REW30303132333435363738393A3B3C3D3E3F60D0<CR><LF>

- Zápis bloku dat na adresu 0x30, data 303132333435363738393A3B3C3D3E3F

\$RER30<CR><LF>

- čtení z adresy 0x30, bude načteno 16 bajtů

Výpis obsahu EEPROM čtecího čipu (EEPROM DUMP)

'\$RED<CR><LF>

Odpověď: čtečka odpoví formátovaným výpisem obsahu EEPROM čtecího obvodu MIFARE. Výpis zahrnuje: manufacturing data + identifikátory HW čtecího čipu a výpis paměti EEPROM. Jedná se o servisní funkci – nedokumentována.

Součástí výpisu je výpis uživatelského prostoru EEPROM, v rozsahu adres 0x30-0x7F. Viz. Zápis/čtení uživatelských dat do paměti EEPROM čtecího obvodu.

Příklad výpisu uživatelského prostoru EEPROM

User	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
03	30	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	FD	BE	03
04	40	AB	CD	EF	12	34	56	12	34	56	78	DD	DF	FF	AA	AC	CC
05	50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
06	60	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	00	00	00	00
07	70	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	00	00	00	00

Uvítáme jakékoliv připomínky a podněty k činnosti systému ACS-line

ESTELAR s.r.o.
Palackého 744/1
769 01 Holešov, Česká republika
tel.: 573 395 466
hotline@estelar.cz