

TECHNICKÝ DOKUMENT

MONET+ B-PROTOKOL

PŘEDKLÁDÁ: VYDAL:	MONET+,a.s. Za Dvorem 505, Zlín – Štípa Zdeněk Skalák David Příbyl Gabriel Bílý
ZPRACOVÁNÍ:	14.03.2022
PLATNOST DO:	<i>Vydání další verze</i>
ČÍSLO VERZE:	1.27.16

Tento dokument zůstává vlastnictvím firmy MONET+,a.s. Duplikace a/nebo předání kterékoli části tohoto dokumentu třetí straně není povoleno bez předchozího písemného souhlasu vedení firmy MONET+,a.s.

1. OBSAH

OBSAH	1
HISTORIE DOKUMENTU	4
SEZNAM POJMŮ / ZKRATEK	7
ÚVOD	8
STRUKTURA ZPRÁVY	9
Hlavička	10
Dodatečná datová pole	12
FID B – Amount 1	13
FID C – Amount 2	13
FID D – Alternate ID	14
FID F – Approval Code	14
FID I/E – Currency code	14
FID J – Card brand	15
FID L – Balance	15
FID P – Primary Account Number	15
FID R – Response Code	15
FID S – Invoice Number	16
FID b – Couvert	17
FID i – Sequence ID	17
FID g – Server message	17
FID I – Sequence Number	17
FID I – Totals/Batch	17
FID m – TOTALS2	18
FID 1 – Card token	18
FID 9 – Customer SubFIDs	18
FID 9P – Partial approval	19
FID 9p – Preauth. data	19
FID 9S – INVOICE NUMBER 2	19
FID 9I – DCC ID	19
FID 9Q – FOR DCC CHECK REQUEST	20
FID 9Q – FOR DCC CHECK REQUEST	20
FID 9D – DCC OFFER FROM SERVER, ON DCC CHECK RESPONSE	20
FID 9G – GASTRO DATA	20
FID 9g – Progress status	21
Servisní příkazy – zprávy	23

Get Application Info (80)	23
Passivate (81)	23
Get Last Transaction (82)	23
Get Last Batch (83)	24
Close Totals (60)	24
Subtotals (65)	25
TMS Call (90)	26
Handshake (95)	26
FIDy ve finančních transakcích	28
Normal purchase (00)	28
Preauthorization (01)	28
Completion of preauthorization (02)	29
Merchandise return (04)	30
Cash advance (05)	31
Reversal (10)	31
Zjištění zůstatku (07)	31
Purchase with cashback (08)	31
PRŮBĚH TRANSAKCE	32
Finanční transakce	32
Vynucené potvrzení (Explicit confirmation)	33
Terminál ve stavu Busy (RC -30)	34
KOMUNIKAČNÍ LINKY	35
UDP / IP / Ethernet	35
TCP / IP / Ethernet – terminál režim KLIENT	35
Víceportový pokladní server	35
Jednoportový pokladní server	35
TCP / IP / Ethernet – terminál režim SERVER	36
PRINTER-LESS TERMINÁL (BEZ TISKÁRNY)	37
Definice zpráv	37
FID 9 – Terminal customer data	37
FID 9t – Terminal ticket information	38
FID 9T – Terminal ticket line	38
Příklad komunikace	38
PODPORA PLATEBNÍCH KARET S PŘÍMOU SLEVOU	39
Dodatečná datová pole	39
FID 9I – Whitelist ID/Version	39
FID 9C – Whitelist items count	39
FID 9N – Whitelist loading Seq. no	40
FID 9W – Whitelist item	40

FID 9V – Whitelist info	40
FID 9E – POS Entry mode	40
FID 9Z – Card token	40
Nové zprávy pro podporu Whitelistu	41
Whitelist info	41
Whitelist Loading	41
Platba s použitím Whitelistu	41
KEEP-ALIVE MECHANISMUS NA KOMUNIKAČNÍ LINCE	43
Keep-alive flag	43
PODPORA 'OFFLINE TRANSAKČÍ' V B-PROTOKOLU	44
N PROTOKOL	45
Get App Info	45
TMS Call	46
PLATEBNÍ APLIKACE	47
PŘÍLOHA 1 : Návrátové kódy serveru	48
Potvrzovací kódy (Approved Codes)	48
PŘÍLOHA 2 : Návrátové kódy terminálů	49
PŘÍLOHA 3 : Příklad průběhu transakce	50
PŘÍLOHA 4 : Vývojové diagramy	53
PŘÍLOHA 5 : Platební transakce s explicitním potvrzováním	57
PŘÍLOHA 6: Faq.	58

2. HISTORIE DOKUMENTU

VERZE	DATUM	AUTOR	POPIS
1.6	05.05.2006	ZSK	Nákup s výběrem hotovosti
1.7	10.05.2010	ZSK	Invoice number I; Currency Code; Sequence ID; Merchant ID
1.8	01.06.2010	ZSK	Blížeší popis komunikace Serial port; UDP/IP network; + příklady
1.9	09.09.2010	ZSK	Upřesnění pojmů délka hlavičky a CRC
1.10	28.03.2010	ZSK	Přidáno: Printer-less terminál; upraveno Currency a Seq-Number
1.11	29.09.2011	ZSK	Přidán GPE kompatibilní mód
1.12	25.01.2012	ZSK	Definováno použití komunikace přes TCP/IP
1.12b	27.01.2012	ZSK	Detekce terminálu při komunikaci po TCP/IP
1.13	06.02.2012	ZSK	Přidán FID 'g'; opraven popis délky FID 'T'
1.14	21.02.2012	ZSK	Přidány další transakční typy 80, 81 and 82
1.14b	13.03.2012	ZSK	Přidána podpora virtuálních terminálů/obchodníků do app. info
1.15	17.04.2012	ZSK	Přidán popis vynuceného potvrzení přes Flags 0x8000
1.15b	27.04.2012	ZSK	Přidána sekce 'PAN-hint'
1.16	18.05.2012	ZSK	Přidána nová služba 83; a FID 'm'; implementován lísteček uzávěrky
1.17	16.08.2013	ZSK	Přidáno pole Invoice 2; echo Invoice(1 2) v odpovědi
1.18	14.01.2014	DPR	Přidán FID 'T' v odpovědi
1.18a	14.02.2014	DPR	Přidána poznámka v sekce 5.4 o pořadí datových polí
1.18b	24.06.2014	DPR	Přidán popis implementace okamžitých slev - whitelist
1.20	04.09.2014	DPR	Přidána kapitola. Keep-alive mechanismus
1.20a	18.09.2014	DPR	Přidána poznámka k sekci 8. GPE kompatibilní mód
1.25	20.02.2015	DPR	Přeskočeny verze s úpravou vztahující se k použití terminálu ve veřejné dopravě. Přidán popis rozšíření funkce parametru FIDu P
1.25.1	16.02.2015	DPR	Přidán popis FID 'J'
1.25.2	19.03.2015	DPR	Přidána operace "Výběr hotovosti" viz Fid T
1.25.3	26.03.2015	DPR	Doplňeny příklady kódů měn
1.25.4	20.04.2015	DPR	Podpora ukládání dat offline transakcí na pokladnu
1.25.5	23.06.2015	APO	Doplnění informace o režimu POS TCP server 7.4, úprava kap. 10
1.25.6	24.07.2015	DPR	Revize nepřesností
1.25.7	25.07.2015	DPR	Upřesnění FIDu I
1.25.8	28.08.2015	DPR	FID 9 G Gastro data, FID L Balance – zůstatek

1.25.9	22.09.2015	DPR	FID F oprava formátu z numerického na alfanumerický
1.25.10	02.10.2015	DPR	Přidání obrázku průběhu transakce
1.25.11	22.10.2015	RPS	Poznámka FID 9G
1.25.12	20.10.2016	DPR	Seznam FIDů u transakcí
1.26.1	12.04.2017	DPR	Popis příkazu TMS Call
1.26.2	12.05.2017	DPR	N protokol
1.26.3	19.05.2017	DPR	FID 9p – Preauth data + Balance inquiry
1.26.4	20.09.2017	DPR	Zjištění zůstatku
1.26.5	06.10.2017	DPR	Oprava překlepů (velikost písmen FIDů)
1.26.6	24.10.2017	DPR	Přidány příklady k hlavičce protokolu a k oddílu 6.2
1.26.7	08.02.2018	DPR	Příloha 5: Platba s explicitním potvrzením
1.26.8	26.06.2018	DPR	Oprava překlepu u příkladu uzávěrky
1.26.9	27.08.2018	GBI	FID D přidány možnosti použití
1.26.10	22.10.2018	ENE	Potlačení hlášení při pasivaci split payment
1.27	06.03.2019	GBI	Převedení dokumentu do Gdocs
1.27.1	19.06.2019	GBI	Přidáno nové pole FID b – Couvert jako součást nové funkcionality pro spropitné
1.27.2	11.07.2019	GBI	Přidáno nové pole FID 9g
1.27.3	28.11.2019	GBI	Rozšířeno pole FID 9g o strukturu dodatkových informací a příklad komunikace V kap. 5.1 přidány flagy: <ul style="list-style-type: none"> • Bit 9 : 0 = neukazuje Progress status, 1 = posílá Progress status • Bit 10: 0 = bez TOKEN, 1 = posílá TOKEN
1.27.4	03.02.2020	GBI	Rozšířeny informace o FID 9g – Progress status (kap. 5.2.28)
1.27.5	24.04.2020	GBI	Opravený překlep na str. 9 v bin čísle
1.27.6	17.06.2020	GBI	Přidány nové operace předautorizace a dokončení předautorizace v kap. 5.4. a 5.2.11 FID T
1.27.7	24.07.2020	GBI	Pole FID 9P – Partial approvement je nyní povinné pro platby/sale a předautorizaci.
1.27.8	15.10.2020	GBI	<ul style="list-style-type: none"> • Odstraněna kapitola 8. GPE kompatibilní mód • Odstraněna kapitola 7.1 Sériový port
1.27.9	01.12.2020	GBI	Doplněna informace ke kapitole 7.3 TCP / IP / Ethernet – terminál režim SERVER
1.27.10	15.01.2021	GBI	Úprava délky pole FID D – Alternate ID
1.27.11	22.03.2021	GBI	Úprava textace v kap. TCP / IP / Ethernet – terminál režim SERVER
1.27.12	26.03.2021	GBI	<ul style="list-style-type: none"> • Úprava délky polí v FID I – Totals/Batch • V kap. "TCP / IP / Ethernet – terminál režim SERVER" doplněno o možnost nevyplňování IP adresy pokladny
1.27.13	08.04.2021	GBI	Změna délky polí u FID B a FID C na max 10 Bytů
1.27.14	16.04.2021	GBI	Kap. 5.3.3. a 7.2. rozšířeny o případ rozpadu TCP spojení
1.27.15	09.03.2022	GBI	<ul style="list-style-type: none"> • Přidána kapitola 6.3 Terminál ve stavu Busy (RC -30)

			<ul style="list-style-type: none"> Přidána informace, že pro aplikaci na platformě Tetra je nastaven keep-alive (ENQ) default timeout 15 vteřin.
1.27.16	14.03.2022	GBI	Opravena informace, že pro aplikaci na všech platformách je nastaven keep-alive (ENQ) default timeout 15 vteřin.

Tab. 1 Verzování dokumentu

3. SEZNAM POJMŮ / ZKRATEK

CAD	Card acceptance device = Terminál = POS
ECR	Electronic cash register = pokladna
FID	Field Identifier of optional Data fields - Identifikátor volitelných polí ve zprávě
POS	Point of sale - place of acceptance of payment cards; also used as a term for the terminal
POSIS	Point of Sale Information System = Authorization server

4. ÚVOD

Tento dokument je určen pro společnosti, které implementují B-protokol pro komunikaci mezi pokladnou a validátorem nebo terminálem pro přijetí karty.

V dokumentu je definováno rozhraní B-protokolu na straně MONET+. Dokument popisuje strukturu zprávy, povolené typy zpráv, pole, které je v nich použito, a obsah přípustný pro tato pole.

Vzhledem k flexibilní povaze zprávy B protokolu musí partneři spolupracovat na transakcích, které mají být podporovány, a volitelných prvcích, které mají být v těchto transakcích použity, které budou specifikovány v samostatném rozšiřujícím dokumentu.

5. STRUKTURA ZPRÁVY

B-protokol je formát používaný jako komunikační protokol mezi pokladnou a terminálem. Hlavním účelem protokolu B je popsat zprávy vyměňované mezi pokladnou a terminálem, definovat všechny přípustné transakce včetně všech datových prvků, které se mohou objevit ve dvojicích požadavků a odpovědí (request a response), a také popsat tok mezi pokladnou a platebním zařízením.

Zpráva je začíná pomocí STX (0x02) a skládá se z hlavičky s pevnou délkou následovanou libovolným počtem volitelných datových polí.

Každé volitelné pole ve zprávě je označeno jedním znakem FID (identifikátor pole), který je přímo před daty. Před každým FID je speciální oddělovač znaků (0x1C). Některá volitelná pole mohou obsahovat dílčí pole (subFid), která jsou označena jedním identifikátorem SFID (identifikátor subFid) umístěným přímo před daty. Před každým SFID je speciální oddělovač znaků (0x1D). Hlavička a data musí být ve formátu ASCII. Konec zprávy je označen speciálním ETX (0x03).

<STX>[Hlavička][Datová část]<ETX>

Následující schéma zobrazuje hlavní položky zprávy.

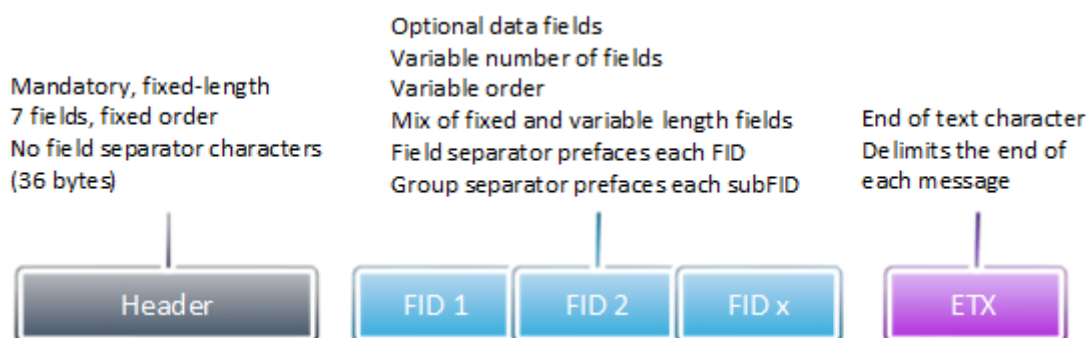


Schéma 1 Struktura zprávy

Každý Fid může být primitivní Fid nebo kompozitní Fid. Primitivní Fid obsahuje Fid Id a hodnotu Fid. Kompozitní Fid obsahuje identifikátor Fid, ale potom pole Sub-Fids. Každý Sub-Fid má předchozí řídicí znak s hexadecimální hodnotou 1D. Každá sub-Fid je zase primitivní fid (viz níže).

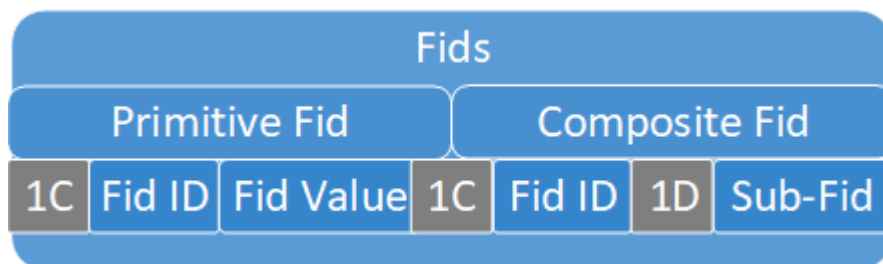


Schéma 2 Formát volitelných polí

Legenda řídicích znaků:

Hex	Symbol	Type	Description
2	STX	Communication Control	Start of Text (začátek zprávy)
3	ETX	Communication Control	End of Text (konec zprávy)
1C	FS	Information Separator	File Separator (oddělovač FIDů)
1D	GS	Information Separator	Group Separator (oddělovač subFIDů)

Tab. 2 Řídící znaky

5.1. Hlavička

Hlavička zprávy je první částí příkazu – nese informace o protokolu, Pos ID, datum a čas transakce, délce datové části a další dodatečné informace v poli Flag.

Struktura hlavičky (pevná délka 36B)

Pole	Formát	Délka (B)	Význam pole	Hodnota
PT	AlfaNumerický	2	Typ protokolu	B0 – BF
PV	AlfaNumerický	2	Verze protokolu	01 – FF
TID	AlfaNumerický	8	Terminal ID	Vysvětleno později
DT	Numerický	12	DateTime Transakce	Aktuální datum a čas
FLG	AlfaNumerický	4	Flag	Vysvětleno později
DLN	AlfaNumerický	4	Délka datové části	0000h – 01FFh
CRC	AlfaNumerický	4	Standard CRC16	Počítaná hodnota

Tab. 3 Header structure

Příklad: [B1 | 01 | A123XX01 | 02 11 25 15 47 21 | 00 00 | 00 00 | A5 A5]

Typ protokolu

Hodnoty v rozsahu "B0" – "BF" (B = protokol pro CADxECD).

- Levý byte B odpovídá identifikaci protokolu protocol ID
- Pravý byte 0 vyhrazený pro ActivityInfoMessage
- Hodnoty 1-F znamenají čísla zpráv (message number)
 - 1 – Transaction Request (pokladna -> terminál)
 - 2 – Transaction Response (terminál -> pokladna)
 - 3 – Ticket Request (pokladna -> terminál)
 - 4 – Ticket Response (terminál -> pokladna)

Verze protokolu

Verze protokolu je číslo v rozsahu "01" – "FF".

Terminal ID

Jedinečný 8byťů dlouhý identifikátor POS.

Datum a čas transakce

Formát YYMMDDHHMMSS = rok|měsíc|den|hodina|min|sec.

Flag

Pole pro přenos dodatečných informací, možný rozsah hodnot: "0000" – "FFFF"

- Bit 0 : 0 = není vyžadován PODPIS, 1 = PODPIS je vyžadován
- Bit 1 : 0 = není LÍSTEČEK, 1 = pokladna musí tisknout LÍSTEČEK
- Bit 2-8 : nevyužity
- Bit 9 : 0 = neukazuje Progress status, 1 = posílá Progress status
- Bit 10: 0 = bez TOKEN, 1 = posílá TOKEN
- Bit 11: 1 = Offline transakce - pokladna může načíst data této transakce
- Bit 12: 1 = Výše placené částky vyžaduje kontrolu totožnosti zákazníka
- Bit 13: 1 = pokladna podporuje Keep-alive mechanismus na TCP spojení
- Bit 14: 1 = při platbě se provede kontrola whitelistu
- Bit 15: 1 = je požadováno potvrzení přijetí zprávy od pokladny

Struktura dodatkových informací:

FLAG_SIGN	0x0001	
FLAG_TICKET	0x0002	
FLAG_EX_PROGRESS	0x0200	// if want B0 with progress status
FLAG_TOKEN	0x0400	// if want TOKEN(family)
FLAG_OFFLINE_DATA	0x0800	// for last offline
FLAG_USER_ID	0x1000	// need user ID
FLAG_KEEP_ALIVE	0x2000	// supports keep alive
FLAG_SPLIT_SALE	0x4000	// split sale support
FLAG_CONFIRM	0x8000	// want confirmation

Délka datové části

Definuje délku datové části příkazu: "00FF" = 255B.

CRC – standard CRC16

Standard CRC16 (check redundancy control): - momentálně nevyužívaná, vyplňuje se konstantou "A5A5".

Příklad reálné hlavičky:

```
0x0000: 02 42 32 30 31 54 31 53 54 30 32 33 30 31 37 31 = '.B201T1ST0230171'
0x0010: 30 32 34 31 35 35 36 34 32 38 30 30 32 30 30 46 = '024155642800200F'
0x0020: 43 41 35 41 35 = 'CA5A5'
```

Flag 8002 = 8002 hex = 1000000000000010 bin

Je tedy nastaven bit 15 a bit 1.

5.2. Dodatečná datová pole

Po hlavičce příkazu je tvořena zbývající část příkazu souborem dodatečných datových polí. Dodatečná datová pole mohou být obsažena v požadavcích od CAD (pokladna) a odpovědích od ECR (terminál). Jednotlivá datová pole jsou definována systémem identifikátorů polí = Field Identifiers (FIDs).

Data příkazu jsou přenášeny v datové části příkazu, kde jsou uložena do jednotlivých datových polí. Každé datové pole je uvozeno znakem - Field Separator FS (1Ch) jak lze vidět níže.

<FS>[field_ID | field_DATA] <FS>[field_ID | field_DATA]

Příklad: <FS>B1000 -> částka = 10.00 Kč

Dodatečná datová pole nemají definované pořadí, jejich vzájemné pořadí může být různé v různých požadavcích i odpovědích a není garantována jeho neměnnost v dalších aktualizacích protokolu.

Parsování odpovědí nemůže být založeno na absolutní pozici, ani na pořadí TAGu. Musí se parsovat každý TAG zvlášť podle <FS> a jména TAGu.

Dodatečná datová pole jsou popsány níže na základě jejich FIDů. Jsou uspořádány do tabulky a seřazeny uspořádání podle abecedy. V tabulce je seznam identifikátorů polí jejich formát, délka (v bytech) a jméno. Zatřítko () v posledních dvou sloupcích tabulky označuje použití tohoto pole v požadavku (RQST) nebo odpovědi (RESP), případně obou typech zpráv.

FID	FORMÁT	DÉLKA (B)	NÁZEV POLE	RQST	RESP
B	PIC 9(8)v99	1-10	Amount 1	✓	
C	PIC 9(8)v99	1-10	Amount 2	✓	
D	PIC 9(2)	1-2	Alternate ID	✓	
F	PIC X(8)	8	Approval code	✓ ¹⁾	✓
I/E	PIC 9(3)	3	Currency code	✓ ²⁾	
J	PIC X(16)	1-16	Card brand		✓
L	PIC 9(16)v99	1-18	Balance		✓
P	PIC 9(19)	13-19	PAN	✓ ³⁾	✓
R	PIC 9(3)	3	Response Code	✓	
S	PIC 9(10)	1-10	Invoice Number 1	✓	✓
T	PIC 9(2)	2	Transaction Type	✓	✓
b	PIC 9(18)	1-18	Couvert	✓	✓
i	PIC 9(9)	9	Sequence ID		✓
g	PIC X(40)	1-40	Server message		✓
l	PIC X(75)	75	Totals/Batch1		✓
m	PIC X(75)	75	Totals/Batch2		✓
1	PIC X(82)	1-82	Card Token		✓

9		variable		✓	✓
9P	PIC 9(1)	1	Partial approvement	✓	
9p	PIC X(90)	4-90	Preauth. data	✓	✓
9S	PIC X(20)	1-20	InvoiceNumber 2	✓	✓
9T	PIC X(43)	1-43	Ticket Line Text	✓	✓
9t	PIC X(1)	1	Ticket Type	✓	✓
9y	PIC X(32)	10-32	Transaction name		✓
9X	PIC H(4)	4	Offset of transaction data	✓	✓
9x	PIC X(400)	1-400	Data of trn. (Base64)		✓
9G	PICX(40)	5-36	Gastro data	✓	✓
9g	PIC H(4)	4	Progress status	✓	✓

Tab. 4 *Tabulka možných fidů*

- 1) Platí pro REVERSAL
- 2) IPro multicurrency terminály
- 3) Ve speciálním případě může být délka i 1 znak viz níže

Kromě Fidů existují i tzv. Subfidy (např.:9S). Formát subfidů se liší od Fidu tak, že je použit znak <GS>.

Příklad: <FS> 9 <GS> S12345678901234567890

Při použití více subfidů v rámci Fidu je možné oddělovat subfidy pomocí <GS> nebo uvést subfid včetně jeho Fidu.

Příklad: <FS> 9 <GS> tline1 <GS> tline2 or <FS> 9 <GS> tline1 <FS> 9 <GS> tline2

Popis významu jednotlivých datových polí je obsažen na následujících stránkách.

5.2.1. FID B – Amount 1

Toto pole obsahuje primární částku. Toto je částka, kterou zákazník bude platit. V případě nákupu s výběrem hotovosti (purchase with cashback transaction) obsahuje součet částky za nákup a částky pro výběr hotovosti.

Požadavek: Povinný pro finanční transakce. Proměnlivá délka 1 až 10 bytů.

Odpověď: Nepovinné. Proměnlivá délka 1 až 10 bytů.

5.2.2. FID C - Amount 2

Toto pole obsahuje sekundární částku. V případě nákupu s výběrem hotovosti, obsahuje toto pole hodnotu hotovosti. Momentálně se toto pole používá pouze pro transakce nákupu s výběrem hotovosti (purchase with cashback transaction).

Požadavek: Toto pole je povinné pro transakce nákupu s výběrem hotovosti.
Proměnlivá délka 1 až 10 bytů.

Odpověď: Nepovinné. Proměnlivá délka 1 až 10 bytů.

5.2.3. FID D – Alternate ID

Pole obsahuje index alternativního Obchodníka (Merchant)/Terminal ID použitého pro tuto transakci. Indexování může probíhat na základě různých parametrů. Alternate ID se může využívat k:

- rozlišování terminálu (pod jedním obchodníkem více terminálů na prodejně),
- rozlišení účelu platby (např. pokuty),
- platbě složenek (pseudo-light virtualizace – virtualizace TID) – hodnota indexu dcery – 1

Požadavek: Nepovinné. Délka 1-2 byte (možná hodnota 0-10; limit virtuálních ID je 10). Požadováno pro finanční transakce na terminálech se zapnutou podporou více obchodníků (sdílené terminály), jinak je toto pole ignorováno.

Odpověď: Nepovinné, pevná délka odpovědi 10 bytů. Formát #:MERCHANTID

5.2.4. FID F – Approval Code

Toto pole obsahuje Autorizační kód reprezentující alfanumerický řetězec generovaný při autorizaci této transakce.

Požadavek: Nepovinné. Pevná délka 8 bytů.

Odpověď: Nepovinné. Pevná délka 8 bytů. Pokud je toto pole obsaženo v požadavku příkazu, je zkopírováno v odpovědi. Pokud toto pole nebylo obsaženo v požadavku příkazu, může být generováno při autorizaci.

5.2.5. FID I/E – Currency code

Toto pole obsahuje kód měny použité během transakce. Z důvodu zpětné kompatibility je podporován FID 'I', ale doporučujeme upřednostnit použití FID 'E'.

Poznámka: Podporované měny jsou závislé na nastavení parametrů terminálu a číslo reprezentující měnu je dle ISO 4217.

Měna	Abecední kód	Číselný kód
Česká koruna	CZK	203
Euro	EUR	978
Americký dolar	USD	840
Libra šterlinků	GBP	826
Polský zlotý	PLN	985
Maďarský forint	HUF	348

Tab. 5 Seznam měn

Ostatní měny naleznete na www.currency-iso.org (ISO4217).

Požadavek: Nepovinné. Pevná délka 3 byty. Povinné v případě finančních transakcí prováděných na terminále s povolenou podporou "multi-currency" jinak ignorováno.

Odpověď: Nepoužito.

5.2.6. FID J – Card brand

Toto pole obsahuje název brandů karty. Nastavení názvů brandů, je závislé na parametrech terminálu.

Požadavek: Nepoužito.

Odpověď: Nepovinné. Proměnlivá délka 1 až 16 bytů.

5.2.7. FID L – Balance

Toto pole obsahuje zůstatek na kartě, vrací se v případě, že je tato funkčnost podporována příslušným autorizačním serverem.

Požadavek: Nepoužito.

Odpověď: Nepovinné. Proměnlivá délka 1 až 18 bytů.

5.2.8. FID P – Primary Account Number

Primary Account number (PAN) – číslo karty

Požadavek: Není použito s výjimkou REFUND (NÁVRATU). Pokud je obsaženo při transakci NÁVRAT slouží jako nápověda pro terminál pro původní číslo karty. Mělo by být ve stejném formátu (maskované), jako původní FID P obdržený v odpovědi transakce terminálu.

Rozšíření: Pokud obsahuje pouze "-" terminál vyzve k ručnímu zadání čísla karty a expirace.

Odpověď: Proměnlivá délka až 19 bytů. Je maskováno dle bezpečnostních standardů.

5.2.9. FID R – Response Code

Pole obsahuje návratový kód z ECR (terminálu) mající význam výsledku transakce. Hodnoty použitých návratových kódů jsou shrnuty v Příloze 1 a Příloze 2.

- Pokud je první znak '-', obsahují následující 2 číslice návratový kód terminálu.
- 3 číslice znamenají návratový kód ze serveru

Požadavek: Nepoužívá se.

Odpověď: Pevná délka 3 byty.

5.2.10. FID S – Invoice Number

Pole obsahuje 'variabilní symbol.

Požadavek: Volitelný pro finanční transakce. Proměnlivá délka od 1 až do 10 bytů. Používáno jako jedinečné označení transakce, generováno obchodníkem. Toto číslo je uvedeno ve výpisu transakcí.

Odpověď: **Opakovaná** hodnota z požadavku.

Poznámka: Invoice 2 má přednost před Invoice 1. Pokud jsou obsaženy v požadavku oba typy Invoice je vrácen pouze Invoice 2.

FID T – Transaction Type

Pole obsahující Typ transakce možné tyto kódy:

Typ transakce	Hodnota
NORMAL PURCHASE (Sale)	00
PREAUTHORIZATION (Předautorizace)	01
COMPLETION OF PREAUTHORIZATION (Dokončení předautorizace)	02
MERCHANDISE RETURN (Refund)	04
CASH ADVANCE	05
BALANCE INQUIRY (Zůstatek)	07
PURCHASE WITH CASHBACK	08
REVERSAL	10
CLOSE TOTALS	60
SUBTOTALS	65
HANDSHAKE	95
GET APP INFO	80
PASSIVATE	81
GET LAST TRANS	82
GET LAST SUMS	83
GET APP CONFIG	84
TMS CALL	90

Tab. 6 Typy transakcí

Požadavek: Pevná délka 2 byty.

Odpověď: Od verze 1.18 je toto pole opakováno i v odpovědi.

5.2.11. FID b – Couvert

Tento FID je určen pro spropitné, ale s tím, že pokladna může poslat i speciální hodnotu '-', čímž indikuje, že podporuje zadání spropitného na terminálu a tudíž v odpovědi i 'navýšenou' částku (FID 'B') a indikaci konkrétní částky spropitného (FID 'b').

Typ **N18** nebo "-". Pokladna může poslat speciální hodnotu '-' čímž indikuje, že se může zadat spropitné na pinpadu a vrátí zpět do pokladny. Pozor, při zadání spropitného se změní FID 'B' celková částka - spropitné je v něm už vždy započítáno (stejně jako cashback amount).

5.2.12. FID i – Sequence ID

Pole obsahuje 'jedinečné' sériové číslo transakce, generované terminálem. Lze použít pro identifikaci transakce. Není zahrnuto do výpisu transakcí!

Požadavek: Nepoužívá se.

Odpověď: Pevná délka 9 bytů. Generováno pro finanční transakce.

Sestává ze tří polí:

- Shift number 9(3) – zvýšeno po každé 1000. uzávěrce
- Batch number 9(3) – zvýšeno po každé uzávěrce.
- Sequence number 9(3) – zvýšeno po každé transakci a nulováno při uzávěrce.

5.2.13. FID g – Server message

Toto pole obsahuje v případě zamítnutí transakce autorizačním serverem dodatečný text objasňující důvod zamítnutí serverem.

Požadavek: Není použito.

Odpověď: Proměnlivá délka 1 až 48 bytes., měl by obsahovat pouze znaky bez diakritiky v případě zpráv ze serveru nebo v ISO 8859-2 v případě lokálních chyb terminálu.

5.2.14. FID I – Sequence Number

The **Sequence Number** help ensure that the host receives and processes every transaction only once.

Požadavek: Volitelný. Pevná délka 10 bytů.

Odpověď: Volitelný. Pevná délka 10 bytů.

5.2.15. FID I – Totals/Batch

Toto pole je tvořeno skupinou polí reprezentující součty terminálu při uzávěrce. Obsahuje shift a batch ID, spolu s počty debitních, creditních operací a součty částek těchto operací. Autorizační server ukládá tyto součty a spolu se svými součty do transakčního logu. Tyto součty obsahují znak znaménka (+ nebo -) na první pozici pole hodnoty.

- Shift number 9(3) – Shift number
- Batch number 9(3) – Batch number
- Number of debits in batch 9(4)
- Amount of debits in batch S9(15)v99 (18 znaků)
- Number of credits in batch 9(4)
- Amount of credits in batch S9(15)v99 (18 znaků)

Příklad "0010010002+000000000000200000001+0000000000000001000"

2	transactions debit (Sale)	200.00
1	transaction credit (Refund)	10.00

Požadavek: Nepoužívá se.

Odpověď: Volitelné. Pevná délka 50 bytů.

5.2.16. FID m – TOTALS2

Toto pole je použito v případě neshody mezi součty terminálu a autorizačního serveru, v tomto případě přenáší toto pole součty terminálu a pole **FID I** přenáší součty serveru. V případě shody součtů se toto pole neposílá.

Požadavek: Nepoužívá se.

Odpověď: Volitelné. Pevná délka 50 bytů. Viz. **FID I**.

5.2.17. FID 1 – Card token

Pole obsahuje Token ke kartě generovaný terminálem

Požadavek: Nepoužívá se.

Odpověď: Vrací se jako odpověď na provedenou platbu

Poznámka: Délka tokenu je kódována do ASCII na 66 znaků.

5.2.18. FID 9 – Customer SubFIDs

Pole Customer SubFIDs je vyhrazeno pro budoucí použití zákazníky. Pokud je použit, bude specifikován v rozšiřující dokumentaci.

Požadavek: Volitelný. Variabilní délka 64 bytů.

Odpověď: Nedá se použít.

5.2.23. FID 9Q – FOR DCC CHECK REQUEST

TYPE(1) TrnCurr(3) TermCountry(3) AppCurr(3) IssCountry(3)
TYPE – requestor type, 'P' for terminal

5.2.24. FID 9Q – FOR DCC CHECK REQUEST

Pro DCC auth request – status for user selection
Pro DCC auth response – status for server processing

STATUS(1):
DCC_STATUS_ACCEPTED = 'A'
DCC_STATUS_REJECTED = 'N'
DCC_STATUS_WAITING = 'W' (přijato, ale čeká na odpověď emitenta issuera)

5.2.25. FID 9D – DCC OFFER FROM SERVER, ON DCC CHECK RESPONSE

TYPE(1),CURRENCY(3),VALUE(1-12),RATE(1-12),LANG(2)
TYPE – 'V' for visa, 'M' for mastercard
VALUE,POINT – first digit – decimal point offset, next is value without the decimal
Example: 52712345 => 27.12345

5.2.26. FID 9G – GASTRO DATA

Pole obsahuje údaje pro platbu stravenkou.

Požadavek:

"x,food,soc" kde:

První znak "x" říká režim kontroly zboží. 0 značí plná autorizace, 1 značí částečnou autorizaci.

Hodnota "food" říká cenu potravin (Food) v celém nákupu v setinách měny.

Hodnota "soc" říká cenu potravin plus cenu sociálních potřeb. Tato hodnota je rezervou pro případ realizace řešení obdobných karet pro sociální dávky a pokud ji některé pokladní systémy zatím nebudou využívat, použijí zde stejnou hodnotu "food".

Logicky vždy platí, že Amount >= Soc >= Food. Pro dnešní nákup v celkové hodnotě 500 CZK na pokladně bez podpory sociálního zboží, kde budou potraviny v hodnotě 300 CZK tedy bude Amount=50000, Soc=30000, Food=30000 a pomocné pole bude 0,30000,30000

Odpověď:

"n,food,soc" kde:

První číslo "n" (int 0-99) říká Index sítě karet. Byla-li použita bankovní karta, bude Index=0. V případě stravenkové karty bude index převzat z číselníku prefixů definovaného bankou.

Hodnota "food" říká částku odečtenou ze stravenkové karty (nebo 0 u bankovní karty)

Hodnota "soc" říká částku odečtenou ze sociální karty (nebo 0 u bankovní karty). Vzhledem k tomu, že jde o funkci RFU a zatím žádnou podporu sociálních karet nemáme, bude zde vždy nula.

Poznámka:

Režim 0 – plná autorizace

V tomto režimu se pro autorizační požadavek se "stravenkovou" kartou kontroluje, zda celý nákup autorizované částky obsahuje pouze potraviny – tedy že Amount = Food a autorizace proti centrále karet proběhne na hodnotu Amount. V případě "sociální" karty se analogicky použije hodnota Soc.

Pokud nebude podmínka splněna, nebo nebude položka Food uvedena, bude transakce zamítnuta s chybou Nepovolené zboží již před provedením autorizace do karetního centra.

Režim 1 – částečná autorizace

V tomto režimu se z karty autorizuje pouze potřebná částka za potraviny či sociální zboží. Hodnota Amount celého nákupu se nekontroluje, a provede se autorizace proti centrále karet přímo na hodnotu Food. V případě "sociální" karty se analogicky použije hodnota Soc.

Pokud bude transakce potvrzena centrem, v odpovědi terminálu se použije nový response kód "částečně potvrzeno = 01" místo "potvrzeno = 00" a pomocné pole bude obsahovat příslušné informace, viz dále. Pole Amount v odpovědi bude obsahovat skutečně autorizovanou částku. Může se tedy, v případě částečné autorizace, lišit od Amount požadavku a nelze ho tedy použít pro párování dotaz-odpověď.

V případě, že pokladna je v režim GASTRO je povinná plnit FID 9G i případě, že je částka food nulová (V nákupu nejsou potravinové položky).

5.2.27. FID 9g – Progress status

Jedná se o příznak toho, co se právě děje (pokud chcete B0 se stavem pokroku). Aby terminál zaslal v B0 progress status, je třeba si o ně říci. Tzn., že v request se v bitové masce nastaví, že pokladna chce extended progress (viz níže).

Struktura dodatkových informací:

FLAG_SIGN	0x0001	
FLAG_TICKET	0x0002	
FLAG_EX_PROGRESS	0x0200	// if want B0 with progress status
FLAG_TOKEN	0x0400	// if want TOKEN(family)
FLAG_OFFLINE_DATA	0x0800	// for last offline
FLAG_USER_ID	0x1000	// need user ID
FLAG_KEEP_ALIVE	0x2000	// supports keep alive
FLAG_SPLIT_SALE	0x4000	// split sale support
FLAG_CONFIRM	0x8000	// want confirmation

Lo-byte - pokrok, ve kterém akce je.

— **Stav** — **Typ operace**

```
Packet 'Send': len=47
```

```
Packet 'Recv': len=46
```

```
Got packet: B0 (len=46)
```

Activity/Confirmation

Extended progress: Flags=0x0200

```
Progress='0100'
```

2019-11-28 08:45:52.353

2019-11-28 08:45:52.353

```
Packet  'Recv': len=46
```

```
0x0000: 02 42 30 30 31 54 31 53 54 30 32 34 36 31 39 31 = '.B001T1ST0246191'
0x0010: 31 32 38 30 38 34 35 35 31 30 32 30 30 30 30 30 = '1280845510200000'
0x0020: 38 41 35 41 35 1c 39 1d 67 30 31 30 33 03      = '8A5A5.9.g0103. '
```

5.3. Servisní příkazy – zprávy

5.3.1. Get Application Info (80)

Slouží k získání identifikace aplikace (hlavička) a verze aplikace (**FID g**). Pokud je terminál definován jako virtuální terminál/obchodník, jsou také vráceny odpovídající hodnoty (opakováním **FID D**).

Může být využit pokladnou pro:

- identifikování terminálu s platební aplikací ČSOB
- zjištění verze aplikace.

Požadavek:

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 32 30 = '.B101 120'
0x0010: 32 32 31 31 35 30 31 33 38 30 30 30 30 30 30 30 = '2211501380000000'
0x0020: 34 41 35 41 35 1c 54 38 30 03 = '4A5A5.T80.'
```

Odpověď:

```
0x0000: 02 42 32 30 31 4c 49 4e 55 58 36 36 36 31 32 30 = '.B201LINUX666120'
0x0010: 33 31 35 30 39 33 33 30 33 30 30 30 30 30 30 34 = '31509330300000004'
0x0020: 41 41 35 41 35 1c 52 30 30 30 1c 67 56 3a 34 2e = 'AA5A5.R000.gV:4.'
0x0030: 31 2e 38 1c 44 31 3a 4c 49 4e 55 58 31 31 31 1c = '1.8.D1:LINUX111.'
0x0040: 44 32 3a 4c 49 4e 55 58 32 32 32 1c 44 33 3a 4c = 'D2:LINUX222.D3:L'
0x0050: 49 4e 55 58 33 33 33 1c 44 34 3a 4c 49 4e 55 58 = 'INUX333.D4:LINUX'
0x0060: 34 34 34 1c 44 35 3a 4c 49 4e 55 58 35 35 35 03 = '444.D5:LINUX555.'
```

5.3.2. Passivate (81)

Pokud byla zahájena platební transakce a terminál je ve stavu, že čeká na vložení platební karty, lze tento příkaz použít k passivaci terminálu. Odpověď na tento požadavek je zpráva **B2** s nastaveným výsledkem operace **R-01** (přerušeno), **R-22** (na terminálu neprobíhala transakce) nebo výsledek transakce (buď potvrzení nebo zamítnutí) v případě, že tento příkaz byl poslán až po zahájení vlastní platby na terminálu.

5.3.3. Get Last Transaction (82)

Tento příkaz umožňuje pokladně opakovaně získat transakční data z **poslední provedené transakce na terminálu**. Odpověď na tento požadavek je stejná jako byl výsledek poslední uskutečněné transakce, to znamená, že může být vrácena i chyba **-22** v případě, že poslední transakce nebyla úspěšná.

Tento příkaz byl vytvořen pro případy, kdy dojde k poruše komunikace mezi pokladnou a terminálem a dává tak možnost pokladně získat data poslední provedené transakce na platebním terminálu, pokladna by si v tomto případě měla porovnat navrácené datové pole **Amount(s)**, **Authorisation Code**, **Sequence ID**, **Invoice** s údaji své poslední transakce a případně provést reversal nadbytečné transakce. V případě úspěšného provedení příkazu `GetLastTransaction` lze také získat data lístečku této transakce – pomocí příkazů **B3/B4**.

Pokud dojde k pádu pokladny, v případě znovu navázání TCP spojení ze strany pokladny nebo terminálů (podle typu připojení client/server) po znovu připojení je potřeba zjistit poslední stav transakce a to pomocí příkazu T82 Get Last Transaction.

Požadavek:

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 32 30 = '.B101          120'
0x0010: 32 32 31 31 35 35 34 31 35 30 30 30 30 30 30 30 = '2211554150000000'
0x0020: 34 41 35 41 35 1c 54 38 32 03 = '4A5A5.T82.'
```

Odpověď: Po úspěšné autorizaci poslední transakce

```
0x0000: 02 42 32 30 31 53 31 41 50 4d 4f 4e 30 31 32 30 = '.B201S1APMON0120'
0x0010: 32 32 31 31 36 35 33 32 37 30 30 30 30 30 30 33 = '2211653270000003'
0x0020: 32 41 35 41 35 1c 52 30 30 30 1c 46 53 50 45 43 = '2A5A5.R000.FSPEC'
0x0030: 49 4d 45 4e 1c 69 30 30 31 30 30 33 30 30 31 1c = 'IMEN.i001003001.'
0x0040: 50 34 37 36 31 2a 2a 2a 2a 2a 2a 2a 2a 30 30 32 = 'P4761*****002'
0x0050: 38 1c 4a 56 49 53 41 03 = '8.JVISA.'
```

Odpověď: V případě neúspěšné autorizace

```
0x0000: 02 42 32 30 31 53 31 41 50 4d 4f 4e 30 31 32 30 = '.B201S1APMON0120'
0x0010: 32 32 31 31 37 30 38 30 37 30 30 30 30 30 30 30 = '2211708070000000'
0x0020: 35 41 35 41 35 1c 52 2d 32 32 03 = '5A5A5.R-22.'
```

5.3.4. Get Last Batch (83)

Tento příkaz slouží k opětovnému získání hodnot poslední uzávěrky terminálu. V případě použití nastavení terminálu v režimu multi-merchant-terminal (terminál je sdílen více obchodníky), lze pomocí datového pole **FID D** získat uzávěrku vztahenou k transakcím, jednoho z virtuálních obchodníků. Lístek uzávěrky se při použití tohoto příkazu na terminálu netiskne.

5.3.5. Close Totals (60)

Tento příkaz slouží k vyvolání k provedení uzávěrky terminálu. Terminál si vyměňuje své součty transakcí s autorizačním serverem a výsledek vrací do pokladny. Zároveň se při uzávěrce posílají na autorizační server offline transakce. A po provedení tohoto příkazu se součty terminálu i autorizačního serveru nulují.

Požadavek:

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 37 30 = '.B101          170'
0x0010: 35 32 39 31 31 33 35 30 36 30 30 30 30 30 30 30 = '5291135060000000'
0x0020: 34 41 35 41 35 1c 54 36 30 03 = '4A5A5.T60.'
```

Potvrzení požadavku:

```
0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 32 33 33 32 37 30 30 30 30 30 30 30 = '5291233270000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Progresy od terminálu (1-N podle délky operace):

```
0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 32 33 33 31 30 30 30 30 30 30 30 30 = '5291233310000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Výsledek operace od terminálu:

```

0x0000: 02 42 32 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B201TJHB0003170'
0x0010: 35 32 39 31 32 33 33 33 31 30 30 30 32 30 30 33 = '5291233310002003'
0x0020: 44 41 35 41 35 1c 52 30 30 30 1c 54 36 30 1c 6c = 'DA5A5.R000.T60.1'
0x0030: 30 30 31 30 34 37 30 30 30 30 2b 30 30 30 30 30 = '0010470000+00000'
0x0040: 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 = '0000000000000000'
0x0050: 2b 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 = '+0000000000000000'
0x0060: 30 30 03                                           = '00.'
  
```

Volitelně potvrzení výsledku pokladnou:

```

0x0000: 02 42 30 30 31 20 20 20 20 20 20 20 20 31 37 30 = '.B001          170'
0x0010: 35 32 39 31 31 33 35 31 30 30 30 30 30 30 30 30 = '5291135100000000'
0x0020: 30 41 35 41 35 03                               = '0A5A5.'
  
```

5.3.6. Subtotals (65)

Tento příkaz slouží k vyvolání k provedení mezisoučtu terminálu. Terminál si vyměňuje své součty transakcí s autorizačním serverem a výsledek vrací do pokladny. Oproti uzávěrce se po provedení tohoto příkazu hodnota součtů nenuluje.

Požadavek:

```

0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 37 30 = '.B101          170'
0x0010: 35 32 39 31 32 30 34 31 36 30 30 30 30 30 30 30 = '5291204160000000'
0x0020: 34 41 35 41 35 1c 54 36 35 03                     = '4A5A5.T65.'
  
```

Potvrzení příjmu požadavku:

```

0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 33 30 32 33 37 30 30 30 30 30 30 30 = '5291302370000000'
0x0020: 30 41 35 41 35 03                               = '0A5A5.'
  
```

Progresy od terminálu (1-N podle délky operace):

```

0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 33 30 32 34 32 30 30 30 30 30 30 30 = '5291302420000000'
0x0020: 30 41 35 41 35 03                               = '0A5A5.'
  
```

Výsledek operace od terminálu:

```

0x0000: 02 42 32 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B201TJHB0003170'
0x0010: 35 32 39 31 33 30 32 34 32 30 30 30 32 30 30 33 = '5291302420002003'
0x0020: 44 41 35 41 35 1c 52 30 30 30 1c 54 36 35 1c 6c = 'DA5A5.R000.T65.1'
0x0030: 30 30 31 30 34 37 30 30 30 30 2b 30 30 30 30 30 = '0010470000+00000'
0x0040: 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 = '0000000000000000'
0x0050: 2b 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 = '+0000000000000000'
0x0060: 30 30 03                                           = '00.'
  
```

Volitelně potvrzení výsledku pokladnou:

```

0x0000: 02 42 30 30 31 20 20 20 20 20 20 20 20 31 37 30 = '.B001          170'
0x0010: 35 32 39 31 32 30 34 32 31 30 30 30 30 30 30 30 = '5291204210000000'
0x0020: 30 41 35 41 35 03                               = '0A5A5.'
  
```

5.3.7. TMS Call (90)

Tento příkaz slouží k vyvolání spojení terminálu na POSMANAGEMENT a terminál, resp. Aplikace MBCA si stáhne aktuální parametry.

Požadavek:

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 37 30 = '.B101          170'
0x0010: 35 32 39 31 31 31 32 34 39 30 30 30 30 30 30 30 = '5291112490000000'
0x0020: 34 41 35 41 35 1c 54 39 30 03                    = '4A5A5.T90.      '
```

Potvrzení požadavku:

```
0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 32 31 31 31 30 30 30 30 30 30 30 30 = '5291211100000000'
0x0020: 30 41 35 41 35 03                                = '0A5A5.          '
```

Progresy z terminálu:

```
0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 32 31 31 31 30 30 30 30 30 30 30 30 = '5291211100000000'
0x0020: 30 41 35 41 35 03                                = '0A5A5.          '
```

Odpověď z terminálu (negativní – chyba spojení)

```
0x0000: 02 42 32 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B201TJHB0003170'
0x0010: 35 32 39 31 32 31 31 32 32 30 30 30 30 30 30 30 = '5291211220000000'
0x0020: 39 41 35 41 35 1c 52 2d 30 36 1c 54 39 30 03 = '9A5A5.R-06.T90.  '
```

Potvrzení požadavku (volitelně)

```
0x0000: 02 42 30 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.B001          170'
0x0010: 35 32 39 31 31 31 33 30 31 30 30 30 30 30 30 30 = '5291113010000000'
0x0020: 30 41 35 41 35 03                                = '0A5A5.          '
```

Příklad úspěšné odpovědi:

Odpověď z terminálu

```
0x0000: 02 42 32 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B201TJHB0003170'
0x0010: 35 32 39 31 32 31 31 32 32 30 30 30 30 30 30 30 = '5291211220000000'
0x0020: 39 41 35 41 35 1c 52 30 30 30 1c 54 39 30 03 = '9A5A5.R000.T90.  '
```

5.3.8. Handshake (95)

Tento příkaz slouží k pro vyvolání testu spojení s autorizačním serverem. Tímto příkazem se ověřuje správnost nastavení adresy autorizačního serveru a komunikační klíče terminálu.

Požadavek:

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.B101          170'
0x0010: 35 32 39 31 30 33 34 34 38 30 30 30 30 30 30 30 = '5291034480000000'
0x0020: 34 41 35 41 35 1c 54 39 35 03                    = '4A5A5.T95.      '
```

Potvrzení požadavku:

```
0x0000: 02 42 30 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B001TJHB0003170'
0x0010: 35 32 39 31 31 33 33 30 39 30 30 30 30 30 30 30 = '5291133090000000'
0x0020: 30 41 35 41 35 03                                = '0A5A5.          '
```

Odpověď terminálu:

```
0x0000: 02 42 32 30 31 54 4a 48 42 30 30 30 33 31 37 30 = '.B201TJHB0003170'  
0x0010: 35 32 39 31 31 33 33 31 35 30 30 30 30 30 31 = '5291133150000001'  
0x0020: 33 41 35 41 35 1c 52 30 30 30 1c 67 41 50 50 52 = '3A5A5.R000.gAPPR'  
0x0030: 4f 56 45 44 1c 54 39 35 03 = 'OVED.T95.'
```

Potvrzení požadavku (volitelně)

```
0x0000: 02 42 30 30 31 20 20 20 20 20 20 20 20 31 37 30 = '.B001 170'  
0x0010: 35 32 39 31 30 33 34 35 34 30 30 30 30 30 30 = '5291034540000000'  
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

5.4. FIDy ve finančních transakcích

Všechny příkazy musí obsahovat FID T – transaction type

5.4.1. Normal purchase (00)

Tento příkaz/transakce slouží k provedení platby.

Povinné FIDy pro požadavek BI: T, B, E¹⁾, D²⁾, 9P.

Volitelné FIDy pro požadavek BI: S, 9S.

FIDy v odpovědi B2: T, P, J³⁾, F, R, g, K, 9p.

Packet 'Send': len=61

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 20 32 30 30 = '.B101          200'
0x0010: 37 32 37 30 39 35 33 35 35 30 30 30 30 30 31 = '7270953550000001'
0x0020: 37 41 35 41 35 1c 54 30 30 1c 42 35 35 30 30 30 = '7A5A5.T00.B55000'
0x0030: 30 30 1c 39 1d 50 31 1c 45 33 34 38 03 = '00.9.P1.E348.  '
```

Packet 'Recv': len=209

```
0x0000: 02 42 32 30 31 48 55 4b 48 30 30 30 31 32 30 30 = '.B201HUKH0001200'
0x0010: 37 32 37 31 31 35 34 31 34 30 30 30 30 30 30 41 = '727115414000000A'
0x0020: 42 41 35 41 35 1c 52 30 31 30 1c 67 70 72 6f 76 = 'BA5A5.R010.gprov'
0x0030: 65 64 65 6e 6f 1c 54 30 30 1c 42 35 35 35 30 39 = 'edeno.T00.B55509'
0x0040: 33 1c 50 32 32 32 32 34 2a 2a 2a 2a 2a 2a 2a 30 = '3.P22224*****0'
0x0050: 31 38 32 1c 4a 4d 41 53 54 45 52 43 41 52 44 1c = '182.JMASTERCARD.'
0x0060: 46 30 30 35 36 33 37 20 20 1c 69 30 30 31 30 31 = 'F005637 .i00101'
0x0070: 37 30 31 38 1c 39 1d 70 41 41 41 42 41 42 45 41 = '7018.9.pAAABABEA'
0x0080: 35 41 66 58 41 6f 49 45 41 77 41 77 4d 44 45 77 = '5AfXAoIEAwAwMDEw'
0x0090: 4d 54 63 77 4d 54 67 41 41 41 41 51 46 50 66 2f = 'MTcwMTgAAAAQFPf/'
0x00a0: 33 54 59 47 42 43 65 57 69 6d 6a 68 36 45 71 71 = '3TYGBCeWimjh6Eqq'
0x00b0: 53 2b 67 48 77 43 6a 71 56 58 67 49 41 41 41 41 = 'S+gHwCjqVXgIAAAA'
0x00c0: 41 41 41 41 41 41 41 41 30 41 6d 59 41 41 3d 3d = 'AAAAAAAAA0AmYAA== '
0x00d0: 03 = '.'
```

5.4.2. Preauthorization (01)

Předautorizace probíhá stejně jako normální platba.

Povinné FIDy pro požadavek BI: T, B, E1), D2), 9P.

Volitelné FIDy pro požadavek BI: S, 9S.

FIDy v odpovědi B2: T, P, J3), F, R, g, K, 9p.

Příklad komunikace:

Packet 'Send': len=69

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 20 32 30 30 = '.B101          200'
0x0010: 34 31 35 31 30 33 33 35 33 30 30 30 30 30 31 = '4151033530000001'
0x0020: 46 41 35 41 35 1c 54 30 31 1c 42 35 35 35 35 1c = 'FA5A5.T01.B5555.'
0x0030: 39 1d 53 4e 41 5a 44 41 52 31 32 33 34 35 36 1c = '9.SNAZDAR123456.'
0x0040: 45 32 30 33 03 = 'E203.  '
```

Packet 'Recv': len=223

```

0x0000: 02 42 32 30 31 54 31 53 54 30 32 34 36 32 30 30 = '.B201T1ST0246200'
0x0010: 34 31 35 31 32 33 34 30 33 30 30 30 32 30 30 42 = '415123403000200B'
0x0020: 39 41 35 41 35 1c 52 30 30 30 1c 67 70 72 6f 76 = '9A5A5.R000.gprov'
0x0030: 65 64 65 6e 6f 1c 54 30 31 1c 42 35 35 35 35 1c = 'edeno.T01.B5555.'
0x0040: 39 1d 53 4e 41 5a 44 41 52 31 32 33 34 35 36 1c = '9.SNAZDAR123456.'
0x0050: 50 35 34 30 30 39 2a 2a 2a 2a 2a 2a 2a 36 39 36 = 'P54009*****696'
0x0060: 33 1c 4a 4d 41 53 54 45 52 43 41 52 44 1c 46 53 = '3.JMASTERCARD.FS'
0x0070: 50 43 4d 4e 30 39 34 1c 69 30 30 31 30 32 34 30 = 'PCMN094.i0010240'
0x0080: 30 38 1c 39 1d 70 49 51 41 42 41 42 67 41 35 41 = '08.9.pIQABABgA5A'
0x0090: 65 66 41 64 49 45 41 77 41 77 4d 44 45 77 4d 6a = 'efAdIEAwAwMDEwMj'
0x00a0: 51 77 4d 44 67 41 41 41 41 51 46 4c 33 6f 33 4c = 'QwMDgAAAAQFL3o3L'
0x00b0: 37 4d 2f 55 56 6f 6b 69 31 64 37 77 71 30 78 75 = '7M/UVoki1d7wq0xu'
0x00c0: 71 2f 77 53 6a 71 73 78 55 41 41 41 41 41 41 41 = 'q/wSjqsxUAAAAAAA'
0x00d0: 41 41 41 41 41 41 67 41 57 41 41 41 3d 3d 03 = 'AAAAAaAWAAA==. '

```

5.4.3. Completion of preauthorization (02)

Při dokončení předautorizace je potřeba zaslat terminálu preauth dat (FID 9p).

Operace T02 vyžaduje použití karty, případně manuální zadání přímo do terminálu za předpokladu, že banka povolí ruční vstup. Z pokladny lze iniciovat pouze za podmínky načtení karty (CL nebo chip), či manuálního zadání čísla karty do terminálu. **Nelze z pokladny provést bez této předchozí činnosti a není možnost udělat kompletní bez přítomnosti karty.** PAN karty je na účtence z předautorizace.

Povinné FIDy pro požadavek B1: T, B, E1), D2), 9p, i.

Volitelné FIDy pro požadavek B1: S, 9S, 9P.

FIDy v odpovědi B2: T, P, J3), F, R, g, K, 9p.

Příklad komunikace:

1) Zakončení s použitím karty:

Packet 'Send': len=198

```

0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 32 30 30 = '.B101          200'
0x0010: 34 31 35 31 30 33 39 32 34 30 30 30 30 30 30 41 = '415103924000000A'
0x0020: 30 41 35 41 35 1c 54 30 32 1c 42 34 34 34 34 1c = '0A5A5.T02.B4444.'
0x0030: 39 1d 53 4e 41 5a 44 41 52 31 32 33 34 35 36 1d = '9.SNAZDAR123456.'
0x0040: 70 49 51 41 42 41 42 67 41 35 41 65 66 41 64 49 = 'pIQABABgA5AefAdI'
0x0050: 45 41 77 41 77 4d 44 45 77 4d 6a 51 77 4d 44 67 = 'EAWAwMDEwMjQwMDg'
0x0060: 41 41 41 41 51 46 4c 33 6f 33 4c 37 4d 2f 55 56 = 'AAAAQFL3o3L7M/UV'
0x0070: 6f 6b 69 31 64 37 77 71 30 78 75 71 2f 77 53 6a = 'oki1d7wq0xuq/wSj'
0x0080: 71 73 78 55 41 41 41 41 41 41 41 41 41 41 41 41 = 'qsxUAAAAAAAAAAAA'
0x0090: 41 67 41 57 41 41 41 3d 3d 1c 69 30 30 31 30 32 = 'AgAWAAA==.i00102'
0x00a0: 34 30 30 38 1c 46 53 50 43 4d 4e 30 39 34 1c 45 = '4008.FSPCMN094.E'
0x00b0: 32 30 33 1c 50 35 34 30 30 39 2a 2a 2a 2a 2a 2a = '203.P54009*****'
0x00c0: 2a 36 39 36 33 03 = '*6963. '

```

Packet 'Recv': len=223

```

0x0000: 02 42 32 30 31 54 31 53 54 30 32 34 36 32 30 30 = '.B201T1ST0246200'
0x0010: 34 31 35 31 32 33 39 34 35 30 30 30 32 30 30 42 = '415123945000200B'
0x0020: 39 41 35 41 35 1c 52 30 30 30 1c 67 70 72 6f 76 = '9A5A5.R000.gprov'

```

```

0x0030: 65 64 65 6e 6f 1c 54 30 32 1c 42 34 34 34 34 1c = 'edeno.T02.B4444.'
0x0040: 39 1d 53 4e 41 5a 44 41 52 31 32 33 34 35 36 1c = '9.SNAZDAR123456.'
0x0050: 50 35 34 30 30 39 2a 2a 2a 2a 2a 2a 2a 36 39 36 = 'P54009*****696'
0x0060: 33 1c 4a 4d 41 53 54 45 52 43 41 52 44 1c 46 53 = '3.JMASTERCARD.FS'
0x0070: 50 43 4d 4e 30 39 34 1c 69 30 30 31 30 32 34 30 = 'PCMN094.i0010240'
0x0080: 30 38 1c 39 1d 70 49 67 41 42 41 42 67 41 35 41 = '08.9.pIgABABgA5A'
0x0090: 65 66 41 64 63 45 4c 51 41 77 4d 44 45 77 4d 6a = 'efAdcELQAwMDEwMj'
0x00a0: 51 77 4d 44 67 41 41 41 41 51 46 4c 33 6f 33 4c = 'QwMDgAAAAQFL3o3L'
0x00b0: 35 45 6b 33 78 67 6b 69 31 64 37 77 71 30 78 75 = '5Ek3xgki1d7wq0xu'
0x00c0: 71 2f 77 53 6a 71 58 42 45 41 41 41 41 41 41 41 = 'q/wSjqXBEBAAAAAAA'
0x00d0: 41 41 41 41 41 41 67 41 57 55 41 41 3d 3d 03 = 'AAAAAAgAWUAA==.'
  
```

2) Zakončení bez přítomnosti karty, PAN se vkládá přímo do terminálu:

Packet 'Send': len=183

```

0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 32 30 30 = '.B101          200'
0x0010: 34 31 35 31 30 34 32 33 39 30 30 30 30 30 39 = '4151042390000009'
0x0020: 31 41 35 41 35 1c 54 30 32 1c 42 34 34 34 34 1c = '1A5A5.T02.B4444.'
0x0030: 39 1d 53 4e 41 5a 44 41 52 31 32 33 34 35 36 1d = '9.SNAZDAR123456.'
0x0040: 70 49 51 41 42 41 42 67 41 35 41 65 66 41 64 49 = 'pIQABABgA5AefAdI'
0x0050: 45 41 77 41 77 4d 44 45 77 4d 6a 51 77 4d 44 67 = 'EAwAwMDEwMjQwMDg'
0x0060: 41 41 41 41 51 46 4c 33 6f 33 4c 37 4d 2f 55 56 = 'AAAAQFL3o3L7M/UV'
0x0070: 6f 6b 69 31 64 37 77 71 30 78 75 71 2f 77 53 6a = 'oki1d7wq0xuq/wSj'
0x0080: 71 73 78 55 41 41 41 41 41 41 41 41 41 41 41 41 = 'qsxUAAAAAAAAAAAA'
0x0090: 41 67 41 57 41 41 41 3d 3d 1c 69 30 30 31 30 32 = 'AgAWAAA==.i00102'
0x00a0: 34 30 30 38 1c 46 53 50 43 4d 4e 30 39 34 1c 45 = '4008.FSPCMN094.E'
0x00b0: 32 30 33 1c 50 2d 03 = '203.P-.'
  
```

Packet 'Recv': len=223

```

0x0000: 02 42 32 30 31 54 31 53 54 30 32 34 36 32 30 30 = '.B201T1ST0246200'
0x0010: 34 31 35 31 32 34 32 35 39 30 30 30 32 30 30 42 = '415124259000200B'
0x0020: 39 41 35 41 35 1c 52 30 30 30 1c 67 70 72 6f 76 = '9A5A5.R000.gprov'
0x0030: 65 64 65 6e 6f 1c 54 30 32 1c 42 34 34 34 34 1c = 'edeno.T02.B4444.'
0x0040: 39 1d 53 4e 41 5a 44 41 52 31 32 33 34 35 36 1c = '9.SNAZDAR123456.'
0x0050: 50 35 34 30 30 39 2a 2a 2a 2a 2a 2a 2a 36 39 36 = 'P54009*****696'
0x0060: 33 1c 4a 4d 41 53 54 45 52 43 41 52 44 1c 46 53 = '3.JMASTERCARD.FS'
0x0070: 50 43 4d 4e 30 39 34 1c 69 30 30 31 30 32 34 30 = 'PCMN094.i0010240'
0x0080: 30 38 1c 39 1d 70 41 67 41 42 41 42 67 41 35 41 = '08.9.pAgABABgA5A'
0x0090: 65 66 41 64 6f 45 4f 77 41 77 4d 44 45 77 4d 6a = 'efAdoEOwAwMDEwMj'
0x00a0: 51 77 4d 44 67 41 41 41 41 51 41 4c 33 6f 33 4c = 'QwMDgAAAAQAL3o3L'
0x00b0: 34 41 41 41 41 41 6b 69 31 64 37 77 71 30 78 75 = '4AAAAAki1d7wq0xu'
0x00c0: 71 2f 77 53 6a 71 58 42 45 41 41 41 41 41 41 41 = 'q/wSjqXBEBAAAAAAA'
0x00d0: 41 41 41 41 41 41 2f 2f 2f 31 41 41 3d 3d 03 = 'AAAAAA//1AA==.'
  
```

5.4.4. Merchandise return (04)

Tento příkaz/transakce slouží k vrácení prostředků na účet zákazníka.

Povinné FIDy pro požadavek B1: T, B, E¹⁾, D²⁾

Volitelné FIDy pro požadavek B1: S, 9S, P, 9P.

FIDy v odpovědi B2: T, P, J³⁾, F, R, g, K, 9p.

5.4.5. **Cash advance (05)**

Tento příkaz/transakce slouží k výběru hotovosti. Typické použití jsou recepce hotelů nebo směnárny.

Povinné FIDy pro požadavek B1: T, B, E¹⁾, D²⁾.

Volitelné FIDy pro požadavek B1: S, 9S, 9P.

FIDy v odpovědi B2: T, P, J³⁾, F, R, g, K, 9p.

5.4.6. **Reversal (10)**

Tento příkaz/transakce slouží ke zrušení poslední podpisové transakce provedené na platebním terminálu, v případě nesouhlasu podpisu. Může být proveden pouze bezprostředně po platební transakci a mezi platební transakcí a jejím reversalem nesmí být provedena uzávěrka.

Povinné FIDy pro požadavek B1: T, F. (Volitelně 9p)

Povinné FIDy v odpovědi B2: T, R, g.

5.4.7. **Zjištění zůstatku (07)**

Tento příkaz/transakce slouží k získání zůstatku na účtu karty. Pokud pokladna pošle FID P, provede se dotaz dle vloženého PAN v opačném případě vyzve terminál zákazníka k vložení karty a číslo karty si načte sám.

Pozn. Při této transakci se neposílá FID B.

Povinné FIDy pro požadavek B1: T (Volitelně P)

FIDy v odpovědi B2: T, L, R, g.

5.4.8. **Purchase with cashback (08)**

Tento příkaz/transakce slouží k provedení platby s výběrem hotovosti.

Povinné FIDy pro požadavek B1: T, B, C, E¹⁾, D²⁾.

Volitelné FIDy pro požadavek B1: S, 9S, 9P

FIDy v odpovědi B2: T, P, J³⁾, F, R, g, K, 9p.

Pozn:

¹⁾ Povinné pro multicurrency terminál.

²⁾ Požadováno pro finanční transakce na terminálech se zapnutou podporou více obchodníků.

³⁾ Toto pole obsahuje zůstatek na kartě, vrací se v případě, že je tato funkčnost podporována příslušným autorizačním serverem.

6. PRŮBĚH TRANSAKCE

6.1. Finanční transakce

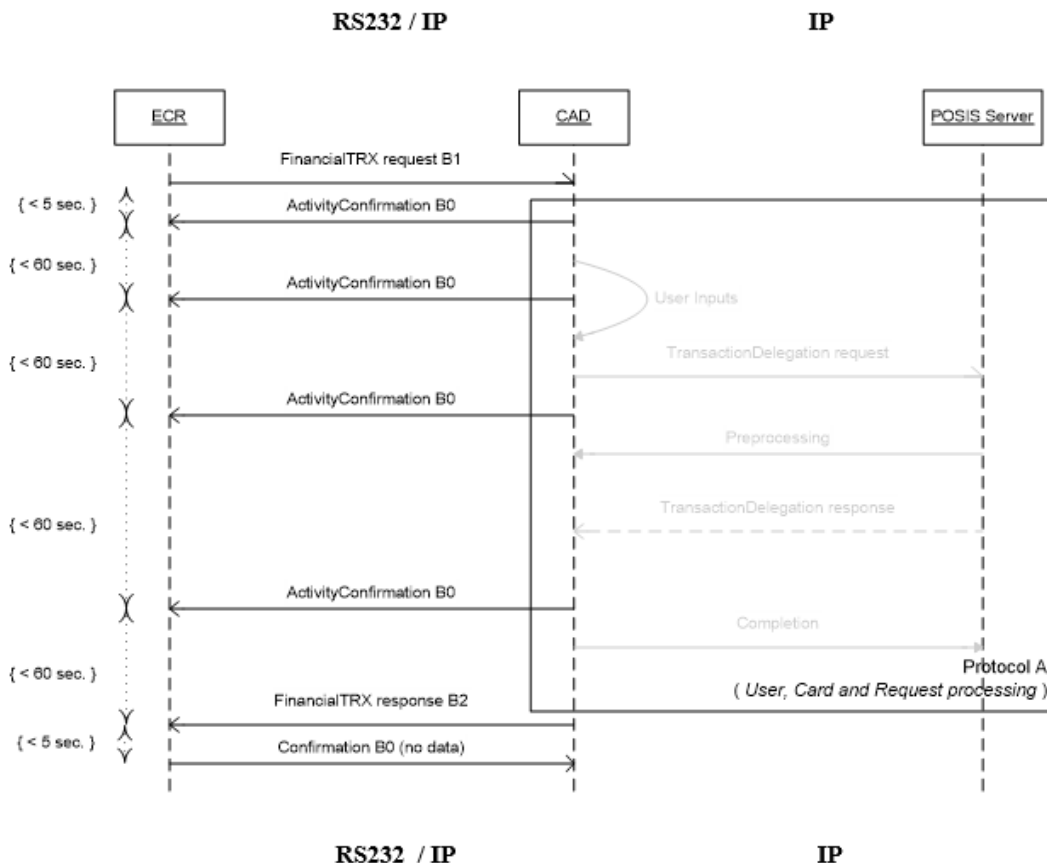


Schéma 3 Tok finančních transakcí

Popis

ECR = Pokladna, CAD = Platební terminál

Krok	Činnost
1	ECR zasílá požadavek/zprávu B1 s finanční transakcí na CAD a čeká po dobu 5s na přijetí potvrzení přijetí této zprávy od CAD.
2	CAD potvrzuje příjem požadavku od ECR zasláním zprávy B0, okamžitě po příjmu požadavku B1.
3	ECR obdržela B0 a vyčkává další zprávy B0 nebo B2. CAD provádí čtení platební karty, zadávání PINU a zahajuje komunikaci s autorizačním serverem. Během těchto činností posílá opakovaně ECR zprávy B0 v časové periodě 60s. (ECR B0 timer = 60s)
4	Autorizace je dokončena, CAD zasílá zprávu B2 do ECR
5	ECR potvrzuje příjem zprávy zasláním B0 do CAD do 5s po obdržení zprávy B2 (timeout <= 5sec).

Tab. 7 Popis toku finančních transakcí

V případě že, pokladna provádí reversal poslední provedené transakce, mělo by součástí tohoto požadavku být pole FID F (authorisation code) nastavený dle výsledku potvrzené transakce.

6.2. Vynucené potvrzení (Explicit confirmation)

Pokladna má možnost nastavit pro terminál mechanismus vynuceného potvrzení. Tento mechanismus funguje tak, že pokud terminál nedostane od pokladny potvrzení přijetí odpovědi B2 s výsledkem autorizace, pomocí zprávy B0, automaticky provádí reversování této transakce. Pro nastavení tohoto mechanismu slouží pokladně bit 15 v části FLAG hlavičky požadavku B1.

Tedy pokud je tento bit nastaven v žádosti B1 (platí pro platby), je toto nastavení zopakováno v odpovědi terminálu B2 pokladny. A terminál očekává do 5s od odeslání odpovědi potvrzení přijetí pokladnou, pokud ho neobdrží provede reversal.

Příklad:

Požadavek pokladny na platbu

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 37 31 = '.B101 171'
0x0010: 30 32 34 31 33 35 36 32 38 38 30 30 30 30 30 31 = '0241356288000001'
0x0020: 39 41 35 41 35 1c 54 30 30 1c 42 31 30 30 1c 39 = '9A5A5.T00.B100.9'
0x0030: 1d 53 41 42 43 44 31 32 33 34 45 46 47 48 03 = '.SABCD1234EFGH. '
```

8000 flag v hlavičce požadavku bit 15 je nastaven – vyžadováno explicitní potvrzení přijetí výsledku platby

Potvrzení přijetí požadavku terminálem

```
0x0000: 02 42 30 30 31 54 31 53 54 30 32 33 30 31 37 31 = '.B001T1ST0230171'
0x0010: 30 32 34 31 35 35 36 32 37 30 30 30 30 30 30 30 = '0241556270000000'
0x0020: 30 41 35 41 35 03 = '0A5A5. '
```

Progresy od terminálu (0..n)

```
0x0000: 02 42 30 30 31 54 31 53 54 30 32 33 30 31 37 31 = '.B001T1ST0230171'
0x0010: 30 32 34 31 35 35 36 32 37 30 30 30 30 30 30 30 = '0241556270000000'
0x0020: 30 41 35 41 35 03 = '0A5A5. '
```

Výsledek platební operace zasílaný terminálem

```
0x0000: 02 42 32 30 31 54 31 53 54 30 32 33 30 31 37 31 = '.B201T1ST0230171'
0x0010: 30 32 34 31 35 35 36 34 32 38 30 30 30 30 30 46 = '024155642800200F'
0x0020: 43 41 35 41 35 1c 52 30 30 35 1c 67 70 72 6f 76 = 'CA5A5.R005.gprov'
0x0030: 65 64 65 6e 6f 1c 54 30 30 1c 42 31 30 30 1c 39 = 'edeno.T00.B100.9'
0x0040: 1d 53 41 42 43 44 31 32 33 34 45 46 47 48 1c 50 = '.SABCD1234EFGH.P'
0x0050: 35 31 36 38 33 2a 2a 2a 2a 2a 2a 2a 35 30 34 34 = '51683*****5044'
0x0060: 1c 4a 4d 41 53 54 45 52 43 41 52 44 1c 46 36 31 = '.JMASTERCARD.F61'
0x0070: 31 30 34 38 20 20 1c 69 30 30 31 30 36 31 30 30 = '1048 .i00106100'
0x0080: 34 1c 31 31 31 34 43 37 46 38 45 39 35 35 30 34 = '4.1114C7F8E95504'
0x0090: 30 33 37 42 44 38 34 41 44 42 43 30 37 39 31 33 = '037BD84ADBC07913'
0x00a0: 37 46 31 31 44 36 42 36 42 46 33 45 33 31 43 43 = '7F11D6B6BF3E31CC'
0x00b0: 35 46 36 36 36 36 35 43 37 41 39 34 41 35 39 46 = '5F66665C7A94A59F'
0x00c0: 34 30 39 36 33 1c 39 1d 70 49 41 41 42 41 44 30 = '40963.9.pIAABAD0'
0x00d0: 41 34 51 63 41 42 42 51 47 4b 67 41 77 4d 44 45 = 'A4QcABBQKqAwMDE'
0x00e0: 77 4e 6a 45 77 4d 44 51 41 41 41 41 51 46 48 35 = 'wNjEwMDQAAAAQFH5'
0x00f0: 37 2f 4c 54 4a 78 47 47 76 59 69 42 75 37 31 71 = '7/LTJxGGvYiBu71q'
```

```

0x0100: 37 6d 75 6a 58 77 69 6a 71 5a 41 41 41 41 41 41 = '7mujXwijqZAAAAAA'
0x0110: 41 41 41 41 41 41 41 41 41 31 77 64 4e 41 41 3d = 'AAAAAAAAA1wdNAA='
0x0120: 3d 03                                     = '=.'

```

8002 flag v hlavičce odpovědi :

bit 15 je nastaven – vyžadováno explicitní potvrzení přijetí výsledku platby

bit 1 je nastaven – pokladna tiskne lísteček

Potvrzení přijetí výsledku platební operace zasílaný pokladnou

```

0x0000: 02 42 30 30 31 54 31 53 54 30 32 33 30 31 37 31 = '.B001T1ST0230171'
0x0010: 30 32 34 31 35 35 36 32 37 30 30 30 30 30 30 30 = '0241556270000000'
0x0020: 30 41 35 41 35 03                                     = '0A5A5.'

```

Tento paket B0 musí být poslán do terminálu jako potvrzení přijetí výsledku platby, do 5s od jeho odeslání, jinak bude terminálem platba reverzována (=zrušena).

Pozn.: Další poznámky k platbě s explicitním potvrzením jsou v příloze 5.

6.3. Terminál ve stavu Busy (RC -30)

Odpověď Busy RC -30 přichází v momentě, kdy terminál není připraven provést požadovanou operaci. Např.: na terminálu právě probíhá platba, uzávěrka, TMS call apod., nebo se terminál nenachází v základním stavu, např. pokud se uživatel pohybuje v menu.

Příklad:

Požadavek na platbu:

Packet 'Send': len=55

```

0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 32 32 30 = '.B101          220'
0x0010: 33 30 33 31 34 30 33 31 33 30 30 30 30 30 31 = '3031403130000001'
0x0020: 31 41 35 41 35 1c 54 30 30 1c 42 31 30 30 1c 45 = '1A5A5.T00.B100.E'
0x0030: 32 30 33 1c 44 31 03                                     = '203.D1.'

```

Odpověď (terminál ve stavu busy):

Packet 'Recv': len=61

```

0x0000: 02 42 32 30 31 54 45 54 52 41 4d 30 31 32 32 30 = '.B201TETRAM01220'
0x0010: 33 30 33 31 35 30 33 31 32 30 30 30 30 30 31 = '3031503120000001'
0x0020: 37 41 35 41 35 1c 54 30 30 1c 52 2d 33 30 1c 67 = '7A5A5.T00.R-30.g'
0x0030: 42 75 73 79 1c 42 31 30 30 1c 44 31 03                                     = 'Busy.B100.D1.'

```

7. KOMUNIKAČNÍ LINKY

7.1. UDP / IP / Ethernet

V případě komunikace pokladny a terminálu prostřednictvím UDP, musí v terminálu být nastaven IP adresa pokladny (nastavení v menu aplikace Manager). A pokladna musí znát adresu terminálu.

Komunikace probíhá prostřednictvím IP protokolu na UDP portu 33333.

Při použití UDP se zasílá naráz kompletní zpráva B protokolu (včetně znaků <STX> a <ETX>) zabalená do jednoho UDP datagramu.

7.2. TCP / IP / Ethernet – terminál režim KLIENT

V případě komunikace pokladny prostřednictvím TCP, musí v terminálu být nastaven IP adresa pokladny (nastavení v menu aplikace Manager). Při tomto způsobu komunikace je pokladna server, který poslouchá na nastaveném portu.

Terminál je klient, který se snaží periodicky (~15 secs) připojit na pokladnu. Po ustavení spojení terminál čeká na přijetí B-Protokolových příkazů. **Protože při tomto způsobu spojení je terminál klient, nemůže provádět udržování spojení!!!** Odpovědnost za udržování spojení je tedy na:

- a) pokladně – serveru použitím TCP keep-alive paketů, nebo
- b) musí být zajištěna konfiguračně pomocí aktivních prvku sítě mezi pokladnou a platebním terminálem, aby byla zajištěna nepřerušitelnost tohoto TCP spojení.

Pokud dojde k pádu pokladny, v případě znovu navázání TCP spojení ze strany pokladny nebo terminálů (podle typu připojení client/server) po znovu připojení je potřeba zjistit poslední stav transakce a to pomocí příkazu T82 Get Last Transaction.

Zprávy B protokolu jsou prostřednictvím TCP přenášeny stejným způsobem jako na sériové lince – jako proud dat.

7.2.1. Víceportový pokladní server

Pokladna naslouchá na specifikované skupině TCP portů, každý port pro jeden terminál. Na serveru je vazba PORT <---> TERMINÁL držena v konfiguraci.

7.2.2. Jednoportový pokladní server

Pokladní server naslouchá pouze na jednom TCP portu. Všechny terminály jsou nakonfigurovány na připojení na tento port. Po vytvoření TCP spojení, může pokladna využít jeden ze dvou způsobů jak identifikovat terminál.

- Pomocí příkazu Get App Info (80)
- Pomocí speciálně upravené B-protokolové žádosti, kdy získá identifikaci platebního terminálu na základě dat odpovědi, konkrétně v hlavičce.

Příklad speciálně upravené žádosti a odpovědi:

Požadavek:

```
0x0000: 02 42 30 30 31 20 20 20 20 20 20 20 20 31 30 30 = '.B101          100'
0x0010: 35 31 33 31 33 32 36 31 31 30 30 30 30 30 30 30 = '51313261100000000'
0x0020: 32 41 35 41 35 1c 54 03 = '2A5A5.5.      '
```

Odpověď:

```
0x0000: 02 42 32 30 31 50 56 54 4d 4f 4e 32 30 31 30 30 = '.B201PVTMON20100'
0x0010: 35 31 33 31 35 32 36 33 30 30 30 30 30 30 30 30 = '51315263000000000'
0x0020: 35 41 35 41 35 1c 52 2D 32 32 03 = '5A5A5.R-22.      '
```

7.3. TCP / IP / Ethernet – terminál režim SERVER

V případě komunikace pokladny prostřednictvím TCP v režimu terminál TCP server by měla v terminálu být nastavena IP adresa pokladny (nastavení v menu aplikace Manager). Případně v aplikaci MCASH. Při tomto způsobu komunikace je terminál server, který poslouchá na nastaveném portu.

Pokud je to chtěné, stačí vyplnit pouze port, bez určení IP (nemusí být žádná IP adresa pokladny vyplněna).

Pokladna se pak chová jako klient, který se snaží periodicky (doporučeno ~15 secs) připojit k terminálu. Po ustavení spojení terminál čeká na přijetí B-Protokolových příkazů.

Terminál dovolí nakonfigurovat více (max 4) pokladen (IP adres) které se můžou připojit.

Při aktivním spojení z pokladny pod IP1, která ustanovila TCP spojení bude terminál komunikovat na tuto IP do doby, než se k terminálu připojí jiná pokladna pod IP2 a ustanoví nové TCP spojení.

Pokud se tak stane během autorizace, může se stát, že request odeslaný z IP1 + stav kdy se připojí IP2 bude odpověď odeslána na pokladnu pod IP2.

Synchronizaci pokladen, tak aby jedna "neshazovala" druhou musí zajistit integrátor. Terminál komunikuje pouze na pokladnu, která je aktuálně připojena, a to i v rámci jedné transakce.

Pro pokrytí případů výpadku spojení či restartu terminálu, např. z důvodu výpadku napájení či problému s kabelem, je nutno podporovat v pokladně aplikační keep-alive.

Protože implementace TCP v OS terminálu nezná linkový keep-alive, je zde podporován aplikační keep alive stejně jako v režimu klient. Pokud dojde např. k výpadku napájení terminálu, pokladna tímto mechanismem problém detekuje, naváže nové spojení a je schopna korektně provést transakci. Detailní popis mechanismu je uveden v kapitole 10.

8. PRINTER-LESS TERMINÁL (BEZ TISKÁRNY)

8.1. Definice zpráv

Pro terminály bez tiskárny jsou lístečky tisknuty na pokladně, navíc je pokladna odpovědná za kontrolu podpisu zákazníka, pokud je tento požadavek zaslán terminálem. V případě nesouhlasu podpisu je pokladna povinná provést REVERSAL transakce prostřednictvím příkazu (typ transakce FID_T='10' s autorizačním kódem v poli FID_F).

Požadavek na kontrolu podpisu zákazníka posílá platební terminál prostřednictvím FLAG v odpovědi B2 výsledku operace, v tomto případě musí dojít ke kontrole podpisu.

Význam FLAG bitů:

- Bit 0: 0 = PODPIS NENÍ vyžadován, 1 = PODPIS vyžadován
- Bit 1: 0 = LÍSTEČEK na terminálu, 1 = Pokladna musí vytisknout lísteček
- Bit 2-15 : not used
- Bit 13: 1 = pokladna podporuje Keep-alive mechanismus na TCP spojení
- Bit 14: 1 = při platbě proběhne kontrola whitelistu
- Bit 15: 1 = je požadováno potvrzení přijetí zprávy od pokladny

Pro podporu tisku lístečku na pokladně jsou definovány následující 2 příkazy/zprávy:

- B3 - Ticket Request (pokladna -> terminál)
- B4 - Ticket Response (terminál -> pokladna)

Krok	Činnost
0	Pokladna dostane odpověď B2 s nastavenou hodnotou FLG:Bit0 na 1
1	Pokladna zašle požadavek na část lístečku - zpráva B3
2	Terminál potvrdí přijetí požadavku B3 zprávou B0
3	Po přijetí zprávy B0 pokladna očekává další zprávy B0 nebo B4
4	Terminál zašle B4 zprávu s částí lístečku do pokladny
5	Pokladna potvrdí přijetí zprávy od terminálu zasláním zprávy B0 (timeout do 5sec).

Tab. 8 *Popis toku transakcí*

8.1.1. FID 9 – Terminal customer data

Datové pole FID 9 jinak nazývané subFID, což znamená, že toto pole nenese samostatně informace, ale slouží jako obálka pro další uživatelská datová pole. V rámci subFID jsou jednotlivá datová podpole odělena znakem Group Separator **GS** (IDh).

8.1.2. FID 9t – Terminal ticket information

Informace o lístečku, pevná délka 1 byte. Z důvodu omezené velikosti bufferů může být lísteček rozdělen na několik částí, předávaných páry zpráv B3/B4.

Požadavek: Pokladna určuje, který lísteček případně část požaduje:

'M' – merchant, je požadována první část lístečku pro obchodníka

'C' – customer, je požadována první část lístečku pro zákazníka

'' – pokračuj další částí lístečku

Odpověď: Platební terminál určuje, jestli lísteček obsahuje ještě další část:

'0' – tato část je poslední

'1' – další data k tisku, pokladna by měla opět provést zprávu B3.

8.1.3. FID 9T – Terminal ticket line

Násobný tag, reprezentující jeden logický řádek tisku. Neměl by obsahovat žádné řídicí znaky diakritika je kódována pomocí ISO-8859-2. Řádky/tagy by měli být vytištěny ve stejném pořadí, jako byly přijaty.

Požadavek: Nepoužívá se.

Odpověď: Různá délka až do 43 znaků.

Struktura dat datového podpole FID 9 T:

- první znak – FONT SELECTOR, určuje velikost fontu jakým má být řádek vytištěn.
- další znaky jsou pak vlastní data řádku.

V současnosti jsou použity 3 různé velikosti písma pro tisk lístečku:

Formát	Použití fontu
0	Font nezměněn – zůstane stávající nastavení
1	12 znaků na řádek
2	24 znaků na řádek
3	42 znaků na řádek

Tab. 9 Formáty použitého písma

8.2. Příklad komunikace

ECR: [STX]B3...[FS]9[GS]tM[ETX]

CAD: [STX]B0...[ETX]

CAD: [STX]B4...[FS]9[GS]t1[GS]T1text1[GS]T2text2[ETX]

ECR: [STX]B0...[ETX]

ECR: [STX]B3...[FS]9[GS]t[SPACE][ETX]

CAD: [STX]B0...[ETX]

CAD: [STX]B4...[FS]9[GS]t0[GS]T3text3[GS]T1end[ETX]

ECR: [STX]B0...[ETX]

9. PODPORA PLATEBNÍCH KARET S PŘÍMOU SLEVOU

Platební karty s přímou slevou jsou takové platební karty, na které při jejich použití obchodník poskytuje slevu. Tyto karty jsou definovány pomocí svého prefixu. Pro využití takových karet umožňuje terminál pokladně nahrát seznam těchto prefixů. Položky tohoto seznamu obsahují typ slevy [1B] a prefix PAN karty [1-7B].

Pro použití této funkcionality jsou nadefinována nová pole pro zprávy B protokolu

FID	Formát	Délka (B)	Název pole	Rqst	Resp
9I	PIC H(8)	8	Whitelist ID/Version	✓	✓
9C	PIC 9	1-4	Whitelist items count	✓	✓
9D	PIC 9(1)	1-2	Whitelist loading Seq. no	✓	✓
9W	PIC X(8)	2-8	Whitelist item	✓	
9V	PIC X(1)	1	Whitelist info		✓
9E	PIC X(1)	1	POS Entry mode		✓
9Z	PIC H(24)	24	Card token		✓

Tab. 10 *Tabulka možných velikostí*

9.1. Dodatečná datová pole

Popis významu jednotlivých datových polí je obsažen na následujících stránkách.

9.1.1. FID 9I – Whitelist ID/Version

Toto pole slouží pokladně pro kontrolu verze aktuálního whitelistu, zapisuje se při vytváření whitelistu a je vrácen při čtení whitelist info. Jeho obsahem je 8 hexadecimálních číslic [8B], tyto čísla nemají pro terminál žádný význam - je to identifikátor pro pokladnu.

Požadavek: Povinný při whitelist load start.

Odpověď: Vrací ho operace whitelist info v případě, že je whitelist definován

9.1.2. FID 9C – Whitelist items count

Počet položek ve Whitelistu, rozsah 1-9999 [1-4B].

Požadavek: Povinný při whitelist load start.

Odpověď: Vrací ho operace whitelist info v případě, že je whitelist definován, a také je vrácen v odpovědi při úspěšném provedení operace whitelist load end

9.1.3. FID 9N – Whitelist loading Seq. no

Číslo aktuální sekvence nahrávání 1–99 [1–2B].

Požadavek: Povinný při whitelist load next.

Odpověď: Vrací ho operace whitelist load next.

9.1.4. FID 9W – Whitelist item

Toto pole obsahuje jednu položku whitelistu, která je tvořena tagem slevy 1B a prefixem karty pro kterou je tato sleva určena. Tento prefix je v rozsahu 1–7 číslic PAN karty. Délka pole 2–8B.

Požadavek: Povinný při whitelist load next.

Odpověď: –

9.1.5. FID 9V – Whitelist info

Toto pole je vráceno v případě rozdělené platby indikuje, že platba byla provedena kartou s nastaveným typem slevy. Tento typ slevy [1B] jedná se o 1. B z položky Whitelistu.

Požadavek: –

Odpověď: Ve výsledku rozdělené platby

9.1.6. FID 9E – POS Entry mode

Pole určuje způsob provádění platby [1B].

- K Ručně zadaná karta
- C Platba magnetickou kartou
- S Čipová karta
- L Bezkontaktní karta

Požadavek: –

Odpověď: Ve výsledku rozdělené platby

9.1.7. FID 9Z – Card token

Token generovaný na základě PAN karty pomocí hashovací funkce 24hexa čísel [24B], slouží pro kontrolu čísla karty při následné rozdělené platbě viz níže.

Požadavek: –

Odpověď: Ve výsledku rozdělené platby

9.2. Nové zprávy pro podporu Whitelistu

Pro nahrávání whitelistu a kontrolu aktuální verze slouží následující zprávy:

T40 – whitelist info
T41 – whitelist load start
T42 – whitelist load next
T43 – whitelist load end

9.3. Whitelist info

Tato zpráva slouží ke zjištění aktuálního stavu whitelistu v terminálu

<- B1: T40
-> B2: T40 R [9I 9C]

9.4. Whitelist Loading

Nahrávání whitelistu se provádí pomocí několika zpráv, zahájí se zprávou *whitelist loading start*,

<- B1: T41 9I 9C
-> B2: T41 R

Pak následují vlastní položky whitelistu pomocí zpráv *whitelist load next*.

<- B1: T42 9N 9W 9W ... 9W (max 15 items)
-> B2: T42 R 9N

A celý přenos je ukončen zprávou *whitelist load end*.

<- B1: T43 9I 9C
-> B2: T43 R 9I 9C

Pokud přenos proběhl správně je starý whitelist nahrazen novým.

9.5. Platba s použitím Whitelistu

Použití whitelistu při vlastní platbě se provádí nastavením HEADER-FLAGS. Pokud je nastavena maska na 0x4000 bude se provádět platba s ohledem na whitelistu v opačném případě je whitelist při platbě ignorován.

Platba s uplatněním okamžité slevy:

<- B1: T00 (Flags | 0x4000 – split sale support) Bxxxxxx
-> B2: T00 (Flags | 0x4000 – split sale transaction) R-99 [9V] 9E 9Z

Při rozdělené platbě (platbě s použitím whitelistu) je vlastní platba po načtení karty přerušena a do pokladny je vrácena odpověď B2, kde je stejně jako v požadavku B1

nastaven Flag | 0x4000 a response kód je -99, pokud prefix karty padne do některého z prefixů whitelistů je vrácen typ slevy v poli FID 9V, každá odpověď obsahuje pole FID 9E – způsob vložení karty a FID 9Z – token vygenerovaný na základě PAN karty.

Dokončení rozdělené platby mimo bezkontaktních karty

Pokladna může v tuto chvíli provést jednu z následujících operací:

- provést pasivaci = zrušit platbu (zaslat do zprávy B1 – passivate T81)
- provést opětovnou platbu (bez použití whitelistu) a to buď na původní částku, nebo částku s uplatněnou slevou.

Pro tyto platby se znovu nenačítá karta nebo zůstává ve čtečce – není nutné provádět kontrolu na číslo karty.

Poznámka: Při zrušení/pasivaci platby se terminálu zobrazí hlášení „transaction aborted“. Pro potlačení tohoto výpisu je potřeba při pasivaci zprávy (T81) v hlavičce B-protokolu) nastavit položku „flag“ na hodnotu ||= 0x4000 (logický součet s hodnotou 0x4000).

Dokončení rozdělené platby bezkontaktní karty

U bezkontaktních karet znamená použití platby se slevou pro zákazníka nutnost přiložit platební kartu 2x. Při prvním přiložení se načte číslo karty a porovná se vůči whitelistu a vrací se příslušná pole. Pokud chce pokladna v tomto kroku přerušit platbu, může opět provést pasivaci nebo dokončit platbu na stejnou částku – pošle se znovu platba bez FLAGS nastavených na 0x4000.

Při uplatnění slevy je nutné po načtení karty, na kterou chceme uplatnit slevu provést pasivaci a znovu zahájit rozdělenou platbou (FLAGS | 0x4000) tentokrát se sníženou částkou (druhé přiložení karty zákazníka), po načtení karty opět dostane pokladna odpověď s tokenem v poli FID 9Z a pokladna musí zkontrolovat, že jedná o stejnou kartu jako při prvním čtení. Pokud jsou tokeny 9Z stejné dokončí platbu na již sníženou částku pomocí platby bez nastaveného FLAGS.

10. KEEP-ALIVE MECHANISMUS NA KOMUNIKAČNÍ LINCE

Keep-alive bude fungovat pouze na terminálech s B-protokolem v TCP terminál. Slouží pro udržení otevřeného TCP kanálu mezi pokladnou a terminálem. Vysílací stranou v tomto případě bude terminál, pokladna odpovídá. Aktivace mechanismu vyžaduje podporu na obou stranách a je popsáno v další kapitole.

Keep-alive se implementuje na linkové vrstvě a sestává s výzvy <ENQ> terminálu a odpovědi <ACK> pokladny. Pokud pokladna neodpoví na <ENQ> do 5 sekund, je linka považována za přerušenu, terminál ukončí dané spojení s pokladnou a dle režimu TCP (klient či server) se buď snaží navázat spojení nové, nebo čeká na obnovení spojení z pokladny.

Terminál bude <ENQ> posílat, pokud v posledních 15 sekundách nebyla na lince žádná komunikace. Jakékoliv přijaté (<STX>, <ACK>, <ENQ>) nebo odeslané (<STX>) data tento timeout restartují.

Poznámka: Pro aplikaci na platformě Tetra je nastaven keep-alive (ENQ) default timeout 15 vteřin (interval mezi ENQ z terminálu), který lze změnit parametrem (např. z TMS nebo v menu aplikace mcash bprotokol).

Je třeba dát pozor, že tyto <ENQ> a <ACK> na lince mohou být proloženy aplikačním B protokol packetem <STX>...<ETX> a nebo <ENQ> z druhé strany komunikace. Tuto situaci je třeba zohlednit, protože může dojít k race-condition (obě strany začnou nezávisle na sobě vysílat):

- pokud terminálu na <ENQ> přijde od pokladny odpověď <STX>...<ETX>, bere se to jako potvrzení keep-alive.
- na následné neočekávané <ACK> terminál nereaguje, ignoruje jej

Keep-alive bude primárně posílat terminál a pokladna odpovídá, nicméně řešení je možné implementovat i v opačném směru. V prozatímní implementaci však nemá praktické využití.

Poznámka: Pokud jsou v cestě mezi pokladnou a terminálem použity aktivní prvky, nerozpozná pokladna tímto mechanismem, zda se jedná o poruchu terminálu nebo aktivního prvku v cestě. V takových případech se doporučuje řešit také dohled funkčnosti aktivních prvků, a to cestou odpovídající typu použitých zařízení.

10.1. Keep-alive flag

Pro zajištění kompatibility se stávajícími řešeními v poli se v B-protokolu použije flag 0x2000 v hlavičce **všech** zpráv, a to následovně:

- a) Pokladna nastavením tohoto flagu (v každé B-protokol zprávě) oznamuje, že podporuje keep-alive.
- b) Pokud od terminálu přijde (na request s nastaveným flagem) odpověď také s nastaveným flagem 0x2000, znamená to, že i terminál podporuje keep alive a zapnul jej.

11. PODPORA 'OFFLINE TRANSAKČÍ' V B-PROTOKOLU

Pro speciální řešení MBCA nově podporuje 'zálohu' offline transakcí na 'pokladnu'. Je pak záležitostí pokladny, jak tyto transakce (pro ni binární bloby) dostane do banky na zpracování (SFTP?).

Pokladna indikuje podporu nastavením bitu 0x0800 ve flag v B1 zprávě pro zahájení transakce.

Pokud terminál tuto vlastnost podporuje a transakce byla offline, indikuje tuto skutečnost nastavením bitu 0x0800 ve flag v B2 odpovědi na transakci.

Pokladna pak může tyto data vyčíst pomocí B5/B6 zpráv.

B5 - request

Supported FID:

- 9X - formát H4; offset for the data in the first request 0000, in another echo of answers

B6 - response:

Podporované FIDy:

- 9Y - formát X32 název transakce, složený z PosID + datum a čas
- 9X - formát H4; offset pro další request, pokud je tento už poslední, tak není přítomen
- 9x - formát X400; BASE64 kódována část binárních dat transakčního blobu

12. N PROTOKOL

Kromě platební aplikace MBCA, která komunikuje pomocí B-protokolu, jsou na terminálu i další funkčnosti/aplikace. Např. aplikace Maintenance, která komunikuje s pokladnou tzv. N-protokolem.

N protokol se liší od B protokolu především písmenem 'N' v hlavičce protokolu oproti původnímu 'B'. Tato změna byla vynucena koexistencí více aplikací na terminálu připojeném k pokladně pomocí rozhraní RS232.

Při komunikaci přes UDP protokol je taky rozdíl v použitém portu, pro N protokol je vyhrazen port 33332.

Protože tento protokol slouží k ovládání aplikace, která se stará o správu aplikací v terminálu,

obsahuje tento protokol pouze dva příkazy:

- GET APP INFO (80)
- TMS CALL (90)

Tyto příkazy mají v N protokolu obdobnou funkci jako výše zmíněné příkazy B protokolu.

GetAppInfo slouží k získání verze aplikace a TmsCall k provedení volání na POSMANAGEMENT server. Při tomto volání může aplikace Maintenance stahovat nové verze aplikací do terminálu a vykonávat update terminálových klíčů.

Pozn: Terminál může obsahovat i jiné aplikace, popisu těchto protokolů se věnují jiné dokumenty.

Příklady použití zpráv:

12.1. Get App Info

Požadavek:

```
0x0000: 02 4e 31 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N101          170'
0x0010: 35 32 39 31 32 31 38 34 34 30 30 30 30 30 30 30 = '5291218440000000'
0x0020: 34 41 35 41 35 1c 54 38 30 03                      = '4A5A5.T80.    '
```

Odpověď terminálu:

```
0x0000: 02 4e 32 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N201          170'
0x0010: 35 32 39 31 33 31 37 30 36 30 30 30 30 30 30 31 = '5291317060000001'
0x0020: 33 41 35 41 35 1c 54 38 30 1c 52 30 30 30 1c 67 = '3A5A5.T80.R000.g'
0x0030: 56 3a 31 2e 32 2e 34 31 03                      = 'V:1.2.41.    '
```

Volitelně potvrzení odpovědi terminálu:

```
0x0000: 02 4e 30 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N001          170'
0x0010: 35 32 39 31 32 31 38 34 35 30 30 30 30 30 30 30 = '5291218450000000'
0x0020: 30 41 35 41 35 03                      = '0A5A5.          '
```

12.2. TMS Call

Požadavek:

```
0x0000: 02 4e 31 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N101          170'
0x0010: 35 32 39 31 32 32 34 32 35 30 30 30 30 30 30 30 = '5291224250000000'
0x0020: 34 41 35 41 35 1c 54 39 30 03 = '4A5A5.T90.  '
```

Potvrzení požadavku:

```
0x0000: 02 4e 30 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N001          170'
0x0010: 35 32 39 31 33 32 32 34 37 30 30 30 30 30 30 30 = '5291322470000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.  '
```

Progresy od terminálu (I-N podle délky operace):

```
0x0000: 02 4e 30 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N001          170'
0x0010: 35 32 39 31 33 32 33 34 36 30 30 30 30 30 30 30 = '5291323460000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.  '
```

Odpověď terminálu (neúspěšné volání):

```
0x0000: 02 4e 32 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N201          170'
0x0010: 35 32 39 31 33 32 33 34 36 30 30 30 30 30 30 30 = '5291323460000000'
0x0020: 39 41 35 41 35 1c 54 39 30 1c 52 2d 31 39 03 = '9A5A5.T90.R-19.  '
```

Volitelně potvrzení odpovědi terminálu:

```
0x0000: 02 4e 30 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N001          170'
0x0010: 35 32 39 31 33 32 33 30 33 30 30 30 30 30 30 30 = '5291323030000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.  '
```

Odpověď terminálu (úspěšné volání):

```
0x0000: 02 4e 32 30 31 20 20 20 20 20 20 20 20 20 31 37 30 = '.N201          170'
0x0010: 35 32 39 31 33 32 33 34 36 30 30 30 30 30 30 30 = '5291323460000000'
0x0020: 39 41 35 41 35 1c 54 39 30 1c 52 30 30 30 03 = '9A5A5.T90.R000.  '
```

13. PLATEBNÍ APLIKACE

Platební aplikace ovládající platební terminál by měla podporovat alespoň minimální množinu příkazů terminálu, pro zajištění úspěšné realizace platby.

Je potřeba podporovat tyto příkazy:

- N-Protocol: TMS Call
- B-Protocol: TMS Call
- B-Protocol: Handshake
- B-Protocol: Normal purchase
- B-Protocol: Reversal
- B-Protocol: Close Totals

Tento výčet je platný pro terminály, kde je tištěn lísteček platební aplikace na tiskárně terminálu. V případě tisku lístečků na tiskárně pokladny je třeba podporovat navíc i příkazy pro vyčtení dat lístečku.

První dva příkazy TMS Call slouží k vyvolání spojení terminálu se serverem posmangmentu a kontroly aktuálnosti verzí aplikací a jejich parametrů, Handshake je diagnostický comand pro kontrolu spojení s autorizačním serverem a zároveň slouží ke kontrole správného nastavení komunikačních klíčů terminálu.

Příkaz Normal Purchase je vlastní provedení platby, kdy po správném naplnění parametrů tohoto příkazu, terminál očekává platební kartu a provede autorizaci požadované platby.

Příkaz Reversal slouží ke zrušení předchozí platby, primárně u plateb, kde je vyžadováno ověření podpisu držitele karty, ale obecně s ním lze zrušit platby do vykonání uzávěrky terminálu.

Příkaz Close Totals, slouží k provedení uzávěrky terminálu a zároveň k zaslání offline transakcí z terminálu na autorizační server.

Popisu jednotlivých příkazů a jejich parametrů Fidů a SubFidů je rozebráno v předchozích kapitolách.

PŘÍLOHA 1 : Návratové kódy serveru

Potvrzovací kódy (Approved Codes)

000 = Approved online // Autorizováno online

001 - 009 = Approved // Autorizováno

010 = Partial approvement // Částečná autorizace, při transakci byla zaplacená nižší částka než bylo požadováno.

Zamítací kódy (Declined Codes)

050 = General // Obecné zamítnutí

051 = Server not connected // Chyba spojení

052 = Number of PIN tries exceeded // Překročen počet pokusů při zadání PIN

053 = Server connected but request error // Připojeno – chyba požadavku

055 = Transaction denied by EMV chip card after 1.decision // Transakce zrušena čipem

056 = Transaction denied by EMV chip card after online authorisation (2.decision) // Trans. zrušena čipem

060 = Merchant cancelation // Zrušeno obchodníkem

061 = Customer cancelation // Zrušeno zákazníkem

062 = Customer declines amount // Zákazník nepotvrdil částku

063 = Timeout cancellation // Zrušeno pro Timeout

070 = Reversal unsuccessful // Neúspěšný reversal

100 = Not allowed transaction // Nepovolená transakce

100 = Unsupported transaction // Nepodporovaná transakce

101 = Wrong card // Neplatná karta

102 = Expired card // Expirovaná karta

103 = Message Data Format Error // Chyba formátu dat

104 = CVC error // Chyba CVC

+ další popis chyb viz manuál ACI SPDH

PŘÍLOHA 2 : Návrátové kódy terminálů

- 01 = MERCH_ERR_USER 1 // normalni ukonceni (uzivatel zrusil castku, PIN,)
- 02 = MERCH_ERR_CANTDOIT 2 // nemohu provest
- 03 = MERCH_ERR_NOPARAM 3 // nejsou definovany parametry karty
- 04 = MERCH_ERR_KOPARAM 4 // chyba v parametrech karty
- 05 = MERCH_ERR_TABLE 5 // obecna chyba tabulky
- 06 = MERCH_ERR_DIAL 6 // nemohu se dovolat
- 07 = MERCH_ERR_UNKNCARD 7 // neznamy typ karty
- 08 = MERCH_ERR_NOTPRIVILEGED 8 // nemam opraveni
- 09 = MERCH_ERR_NOTVALID 9 // chyba karty (spatny PAN len, luhn, service)
- 10 = MERCH_ERR_EXPIRED 10 // prosla
- 11 = MERCH_ERR_TOOYOUNG 11 // jeste neplati
- 12 = MERCH_ERR_TOOLITTLE 12 // prilis maly obnos -> pouzijte cash
- 13 = MERCH_ERR_TOOBIG 13 // prilis velky obnos, pouzijte !?!?!?
- 14 = MERCH_ERR_SAMEPAN 14 // dve operace se stejnou kartou po sobe
- 15 = MERCH_ERR_SERVICE 15 // chyba servisniho kodu
- 16 = MERCH_ERR_PINPAD 16 // chyba pinpadu
- 17 = MERCH_ERR_TRANS_KEY 17 // chyba transportniho klice
- 18 = MERCH_ERR_TIMEOUT 18 // run out of time

- 19 = MERCH_ERR_AUTH 19 // chyba autorizace
- 20 = MERCH_ERR_FULL 20 // no more place to make transaction ...
- 21 = MERCH_ERR_INTERN 21 // internal, unspecified error
- 22 = MERCH_ERR_MASTER 22 // error in datas supplier by master, cash register,
- ...
- 23 = MERCH_ERR_SIGN 23 // signs don't match

- 24 = MERCH_ERR_EMV 24 // EMV card & not supported callback
- 25 = MERCH_ERR_ARPC 25 // EMV card rejected, after online on 2nd
GENERATE AC // command
- 29 = MERCH_ERR_BLOCK 29 // EMV card blocked !!!!

- 99= Special, split-sale intermediate result 99 // Rozdělená platba - uplatnění okamžité slevy

Pozn:

- 25 je lokální chyba, která se na 99% vyskytuje při platbách čipovými MasterCardem do modelovky.
- Server nepošle kryptogram, a proto to karta reverzuje platební transakci.

PŘÍLOHA 3 : Příklad průběhu transakce

- X - řídicí znaky
- X - proměnlivé znaky v hlavičce
- X - FID identifikátory

Packet 'Send': len=55

```
0x0000: 02 42 31 30 31 20 20 20 20 20 20 20 20 31 30 30 = '.B101100'
0x0010: 35 31 33 31 33 32 36 31 31 30 30 30 30 30 30 31 = '5131326110000001'
0x0020: 31 41 35 41 35 1c 54 30 30 1c 42 31 30 30 1c 45 = '1A5A5.T00.B100.E'
0x0030: 32 30 33 1c 44 31 03 = '203.D1.'
```

BxSend(): send() -> 55

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 30 30 30 30 30 30 30 30 = '5131526100000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 30 30 30 30 30 30 30 30 = '5131526100000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 31 30 30 30 30 30 30 30 = '5131526110000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 36 30 30 30 30 30 30 30 = '5131526160000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 37 30 30 30 30 30 30 30 = '5131526170000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 38 30 30 30 30 30 30 30 = '5131526180000000'
```

```

0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 31 39 30 30 30 30 30 30 30 = '5131526190000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 30 30 30 30 30 30 30 30 = '5131526200000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 31 30 30 30 30 30 30 30 = '5131526210000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 31 30 30 30 30 30 30 30 = '5131526210000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 31 30 30 30 30 30 30 30 = '5131526210000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 31 30 30 30 30 30 30 30 = '5131526210000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

Packet 'Recv': len=38
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 34 30 30 30 30 30 30 30 = '5131526240000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.
Got packet: B0
Activity/Confirmation

```

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 36 30 30 30 30 30 30 30 = '5131526260000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 36 30 30 30 30 30 30 30 = '5131526260000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 37 30 30 30 30 30 30 30 = '5131526270000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=38

```
0x0000: 02 42 30 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B001PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 32 37 30 30 30 30 30 30 30 = '5131526270000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

Got packet: B0

Activity/Confirmation

Packet 'Recv': len=61

```
0x0000: 02 42 32 30 31 50 56 54 4d 4f 4e 32 00 31 30 30 = '.B201PVTMON2.100'
0x0010: 35 31 33 31 35 32 36 33 30 30 30 30 30 30 30 31 = '5131526300000001'
0x0020: 37 41 35 41 35 1c 52 31 35 30 1c 50 2a 2a 2a 2a = '7A5A5.R150.P****'
0x0030: 2a 2a 2a 2a 2a 2a 2a 2a 30 35 33 34 03 = '*****0534.'
```

Got packet: B2

Response

Payment: RC='150', Auth=", PAN='*****0534', SeqID="

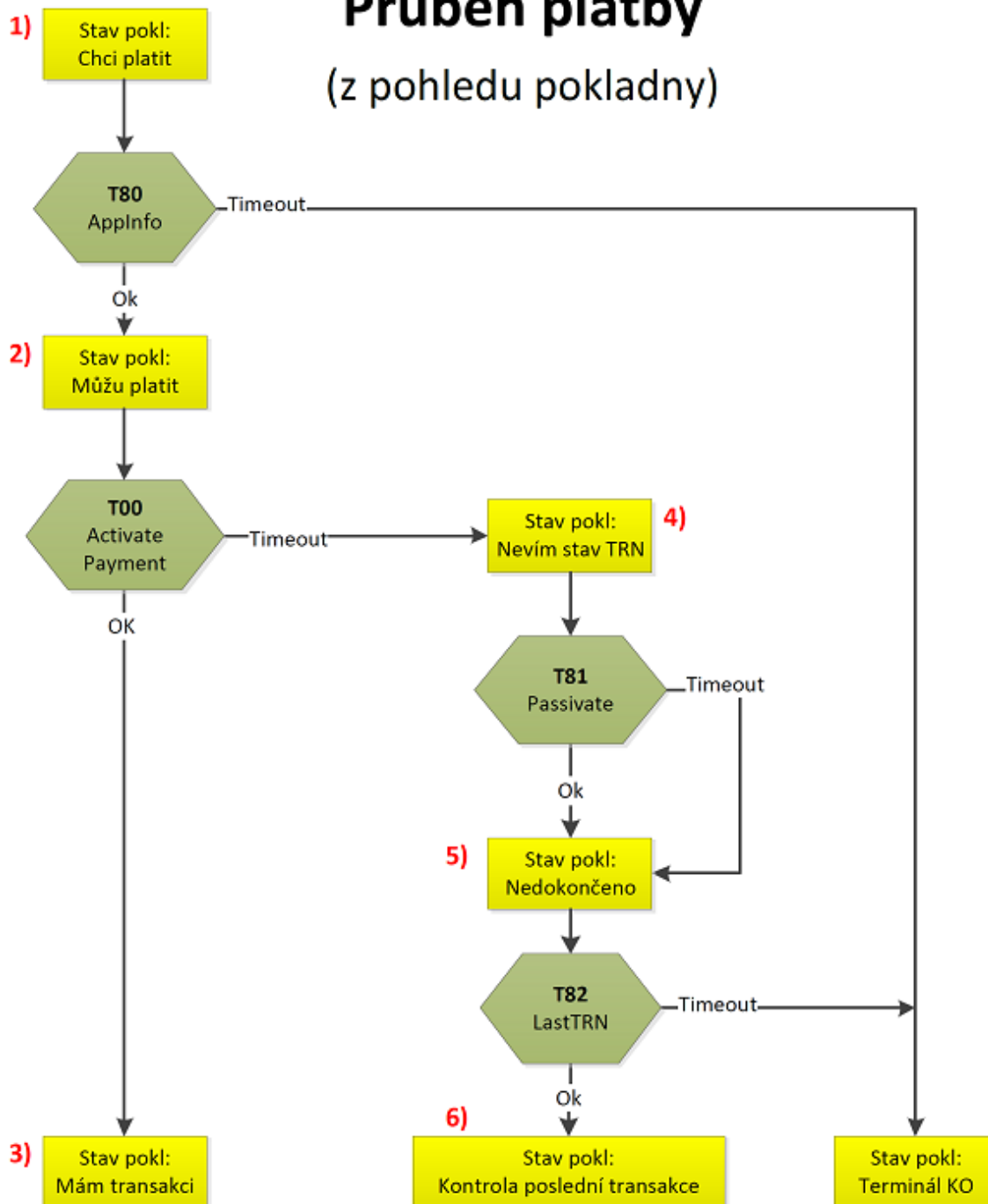
Packet 'Send': len=38

```
0x0000: 02 42 30 30 31 20 20 20 20 20 20 20 20 31 30 30 = '.B001 100'
0x0010: 35 31 33 31 33 32 36 31 31 30 30 30 30 30 30 31 = '5131326290000000'
0x0020: 30 41 35 41 35 03 = '0A5A5.'
```

BxSend(): send() -> 38

PŘÍLOHA 4 : Vývojové diagramy

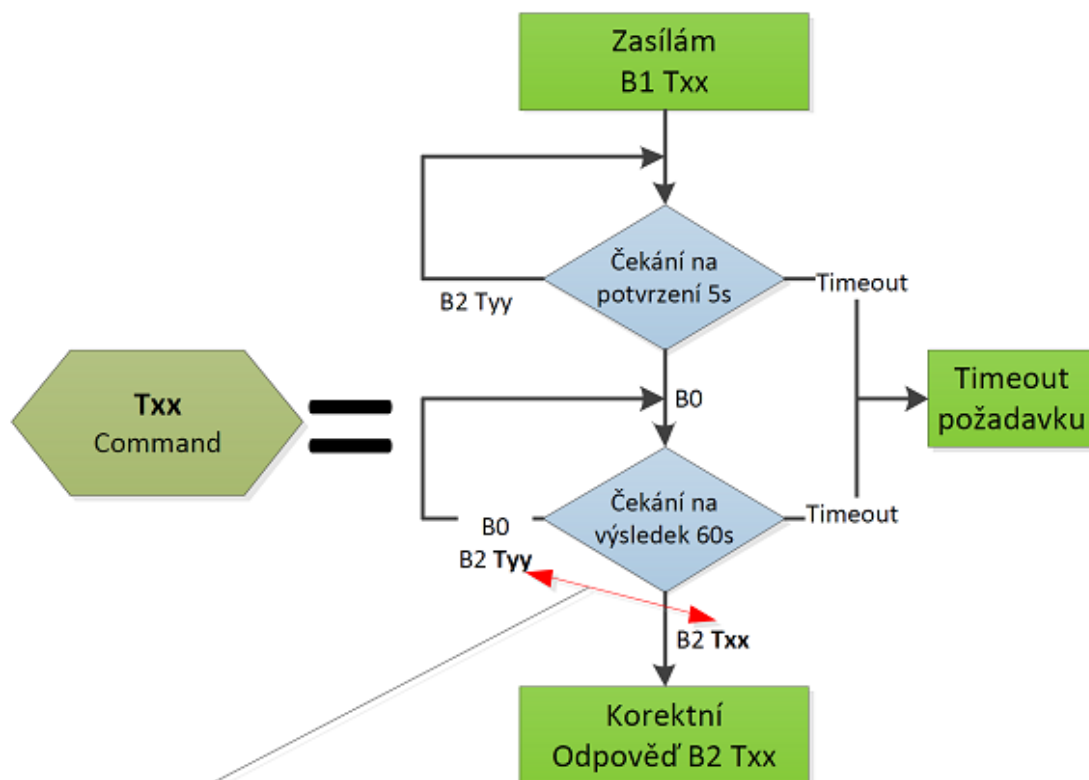
Průběh platby (z pohledu pokladny)



Legenda obrázku

- 1) Terminál je v klidovém stavu, pokladna má připravenou platbu. Jako první krok prověří pokladna stav terminálu. Funkce AppInfo (command T80, viz dokumentace protokolu) nevrací v tomto okamžiku žádný významný výstup, pouze ověřuje stav terminálu. V případě jejího selhání je terminál v nekorektním stavu a je nutno provést kontrolu připojení terminálu. Nelze iniciovat platbu na terminálu.
- 2) Terminál odpověděl, pokladna tedy připraví operaci platby a pošle ji do terminálu (command T00, data commandu viz dokumentace). Pokud při provedení operace platba nedostane pokladna korektní odpověď neví, v jakém stavu platba je. Příkaz nemusel terminálu dorazit, ale také mohla být platba potvrzena a ztratila se odpověď. Proto se provede zjištění stavu terminálu a poslední transakce.
- 3) V případě korektní odpovědi je transakce Ok, ukládá se a pokladna se zachová podle ResponseCode v transakci a stejně tak kontroluje variabilní symbol a částku v odpovědi.
- 4) Po nekorektní odpovědi na platbu provede pokladna preventivně Passivete na terminálu (command T81, viz dokumentace protokolu) aby zajistila, že terminál necykluje např. ve smyčce čekání na protažení karty, zákaznický vstup a podobně.
- 5) Následně zavolá operaci LastTrn (command T82, viz dokumentace protokolu), čímž zjistí, jakou transakci terminál naposledy zpracovával a jak dopadla. Pokud zde nastane timeout, jde o poruchu komunikační linky s terminálem a je třeba prověřit jeho stav.
- 6) Data poslední TRN terminálu jsou k dispozici. Pokladna je porovná se svou aktuálně řešenou transakcí. Možnosti jsou dvě:
 - a) VS (a částka) se shoduje - terminál tedy obdržel požadavek a transakci provedl. Výsledek je tedy v Response Code (operace T82 jej vrací, viz dokumentace) a pokladna se podle něj zachová jako při korektním průběhu komunikace v bodě 3.
 - b) VS (a částka) se neshoduje - terminál tedy jako poslední prováděl starší transakci a aktuálně řešený požadavek pokladny mu vůbec nedorazil. Pak tedy žádná platba nebyla provedena, a je na pokladně, zda pokus o zaplacení aktuální transakce bude znovu opakovat

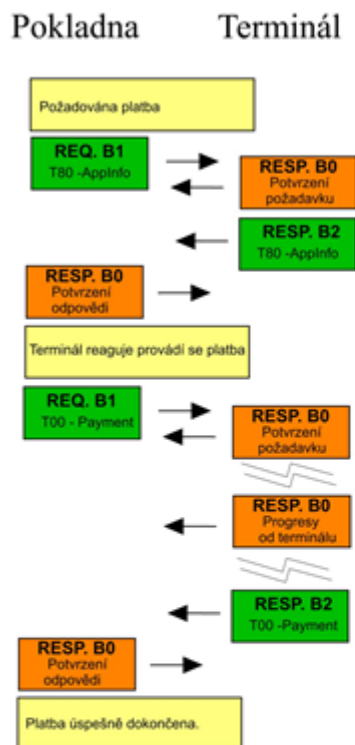
Jeden krok platby (detail komunikace)



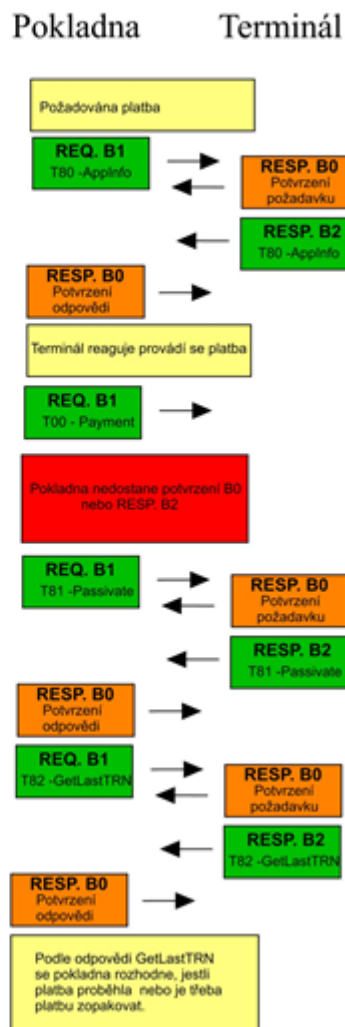
Poslední úprava Bprot:

V odpovědi B2 je vždy Txx
Podle něj poznáme, zda jde
o korektní odpověď Txx nebo
starý responze jiné operace Tyy

Normální průběh platby



Platba při výpadcích komunikace

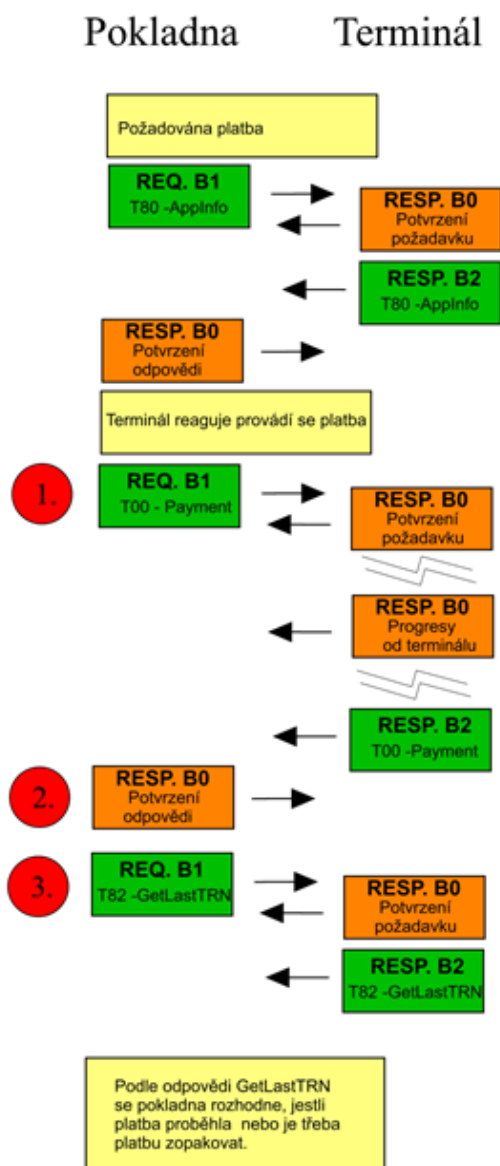


PŘÍLOHA 5 : Platební transakce s explicitním potvrzováním

Mechanismus explicitního potvrzování platby zajišťuje automatické zrušení platby (Reversal) v případě, že dojde ke ztrátě konektivity mezi platebním terminálem a pokladnou. Zároveň však může způsobit automatické zrušení i v případě, kdy se pokladně nepodaří doručit potvrzení do 5s od odeslání výsledku platby.

Při použití explicitního potvrzování, proto doporučujeme po zaslání potvrzení výsledku platby pokladnou, provést příkaz T82 Get Last transaction a porovnat jestli odpověď na T82 je shodná s předcházejícím výsledkem platby, aby se ověřilo, že terminál obdržel potvrzení platby včas a platba tedy nebyla reversována.

Platba s explicitním potvrzením



Pozn.:

1. Platba s požadavkem explicitního potvrzení
2. Potvrzení přijetí výsledku platby pokladnou
3. Ověření doručení potvrzení, pomocí zjištění poslední provedené transakce

PŘÍLOHA 6: Faq.

Otázky k B protokolu:

1. *Jak funguje Keep Alive mechanismus v případě nastavení komunikační linky TCI/IP/Ethernet – terminál režim SERVER, z textu není zřejmé co je "B-protocol v TCP terminal" viz popis v kapitole Keep-alive mechanismus?*

Nastavení režimu

- TCP / IP / Ethernet – terminál režim KLIENT
- TCP / IP / Ethernet – terminál režim SERVER

Má vliv na způsob TCP připojení terminálu a pokladny, které z těchto zařízení zahajuje TCP spojení.

Co se týče Keep-alive mechanismu je v obou režimech stejný, po jeho nastartování, třeba příkazem GET APP INFO s nastavenými flag 0x2000, začne terminal posílat znak <ENQ>, na který očekává odpověď od pokladny znak <ACK> nebo ACK nahrazuje příkaz od pokladny.

Viz kapitola 10. Keep-alive.

2. *V kapitole 6 Průběh transakce je sekvenční diagram, popisující provedení platební transakce. Liší komunikace v případě režimu KLIENT a režimu SERVER?*

Při všech způsobech připojení pokladny a terminálu (sériovém, UDP, TCP-Server, TCP-Klient), je v B-protokolu master pokladna a terminál slave tj. pokladna posílá požadavky a terminál odpovídá.

3. *Jde odvolat platbu příkazem T81 PASSIVATE před potvrzením zprávou B2 vždy, nebo může terminál příkaz odmítnout (např. v bance už bylo hotovo)?*

Poslat jde vždy, ale nelze pasivovat rozběhlou komunikaci s autorizačním serverem. Tzn., že pokud pošlete PASSIVATE při čekání na kartu nebo PIN, transakce se přeruší, pokud by terminál dostal PASSIVATE až v průběhu komunikace s autorizačním serverem – platební transakce se nepřeruší a dostanete výsledek platební transakce.

4. *Týká se transakce T10 REVERSAL pouze poslední provedené transakce? Jestliže ne tak do kdy (za jakých podmínek) lze REVERSAL provést?*

Ano, REVERSAL lze provést jen na poslední platební transakci a to do provedení uzávěrky. Tento příkaz je určen pro rušení platební transakce pro případ nesouhlasu podpisu nebo nevytištění lístečků o platební transakci, takže se spíše předpokládá jeho použití bezprostředně po platební transakci.

5. *Co je transakce typu T05 CASH ADVANCE?*

Výběr hotovosti – hotely, směnárny.

6. *K čemu je transakce typu T95 HANDSHAKE?*

Pro ověření spojení terminálu s autorizačním serverem.

7. *Co vrací transakce typu T65 SUBTOTALS?*

Vyměňuje součty s autorizačním serverem, jako při uzávěrce (T60 CLOSE TOTALS), ale nenuluje je.

8. *Transakce typu T60 CLOSE TOTALS vrací hodnoty z poslední uzávěrky spuštěné manuálně na terminálu, nebo tuto uzávěrku iniciuje?*

T60 CLOSE TOTALS – provádí uzávěrku.

9. *Předpokládáme, že pokladna musí být po celou dobu probíhající platby připojena k terminálu. Nebo stačí jen zaúkolovat a posléze se zeptat na výsledek?*

Ano, spojení mezi pokladnou a terminálem by nemělo být přerušované. Pokud nedostane odpověď na autorizaci, existuje mechanismus na zjištění poslední odpovědi (GetLastTransaction), ale ten spíš určen k řešení havarijních stavů.