



FMS OEM CHIP V5+

Obsah:

ZÁKLADNÍ POPIS	2
INTEGRACE FMS CHIPU DO SYSTÉMU	3
INTERFACE DO NADŘAZENÉHO SYSTÉMU	3
SPI INTERFACE	5
Příkaz SETTINGS na SPI	8
Příkaz READ na SPI	8
READ: FIRMWARE	8
READ: CAN BUS DATA	9
READ: TACHOGRAPH DATA	13
READ: SETTINGS	14
READ: SETTINGS2	15
READ OBD DTC TABLE	15
SEND CAN USR MSG	16
SET CAN USR RCV MSG	16
READ CAN USR RCV MSG	16
RS232 INTERFACE	17
Příkaz SETTINGS u nejčastěji monitorovaných vozidel.	23
Čtení chybových kódů vozidla	25
POWER CONTROL	26
SPECIFIKACE OBJEDNÁVKY	26
UPOZORNĚNÍ	26

Ing. David Španěl

Mgr. Vítězslav Rejda

CANLAB s.r.o.

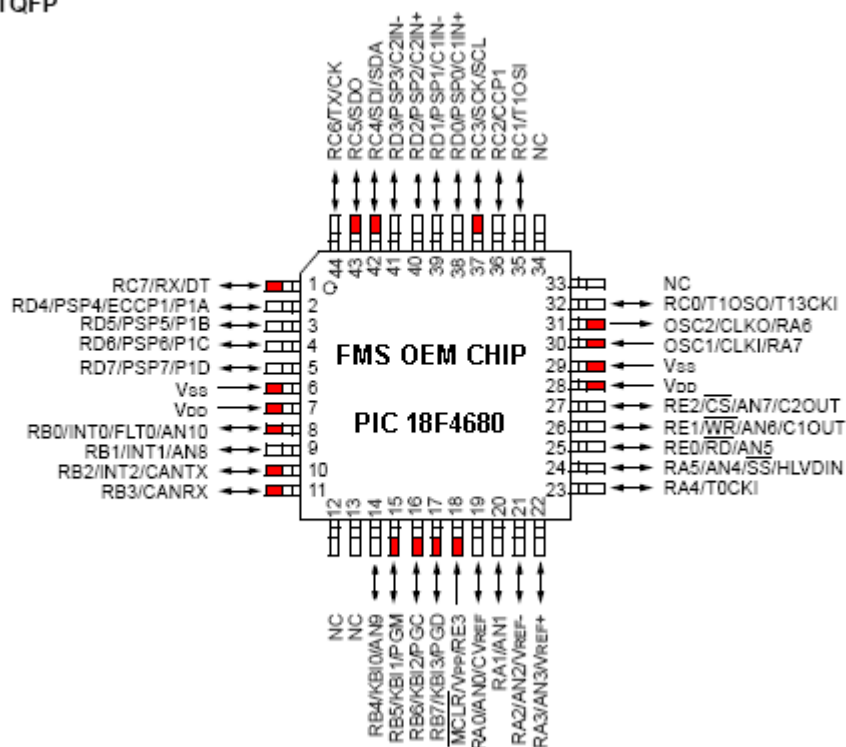
Základní popis

FMS OEM CHIP je předprogramovaný microcontroller PIC 18F4680 v pouzdře 44-pin TQFP, určený k integraci do systému pro sledování vozidel, systémů pro správu vozového parku apod. Chip provádí předzpracování dat z vozidlové sběrnice CAN a případně i z digitálního tachografu DTCO 1381, ke kterému je připojen prostřednictvím jeho info interface. Současná verze firmware podporuje zpracování dat ze sběrnice CAN u vozidel, která využívají protokolu SAE J1939 a dále zpracování dat ze sběrnice CAN u části vozidel koncernu VW (včetně Škoda), Mercedes Sprinter, Nissan atd.. Tato předzpracovaná data poskytuje do nadřazeného systému prostřednictvím sběrnice SPI pracuje v režimu SPI slave nebo RS232 (UART).

Verze 5 tohoto chipu dovoluje číst chybové kódy vozidla a taktéž obsahuje podporu pro přímé uživatelské čtení dat z CANu nebo zasílání dat na CAN.

Do firmware lze na přání zákazníka zapracovat jiný vůz vybavený sběrnici CAN. Podmínkou je krátké zapůjčení tohoto vozidla pro sérii měření, jejichž úkolem je identifikovat požadovaná data na sběrnici CAN tohoto vozidla.

44-Pin TQFP



Obr. 1: Pouzdro FMS OEM CHIPu

Integrace FMS chipu do systému

Obvod vyžaduje pro svoji činnost napájecí napětí 5V a zdroj hodinového kmitočtu 10MHz. Aby byla možné provádět update firmware obvodu, je doporučeno vybavit jej programovacím konektorem.

Dále je vhodné při návrhu počítat s volitelným zapnutím/vypnutím zakončovacího odporu CAN sběrnice o velikosti 120 ohmu. Tento zakončovací odpor se využívá při připojení na tzv. FMS bránu u nákladních vozidel. CAN bus je obvykle zakončen na každé straně zakončovacím odporem 120 ohmu (mezi CAN H a CAN L se naměří odpor 60 ohmu).

Signál MCLR je vhodné zapojit jako externí reset obvodu z nadřazeného výstupu. Signál SDO je v době, kdy je CS neaktivní přepnut jako vstup.

K chipu je možné připojit 2 indikační LED. LED připojená k pinu RC0 indikuje činnost CANu, při příjmu zprávy, která nese data o rychlosti vozidla, výstup změní stav. LED připojená k pinu RC1 pak indikuje činnost tachografu. Při příjmu celé validní zprávy z tachografu výstup změní stav.

Rozhraní CAN je po zapnutí aktivní nebo neaktivní, je li požadováno v objednávce aktivní rozhraní bez nutnosti jeho inicializace, je tuto funkčnost třeba specifikovat v objednávce, pak je nastavení uloženo v interní EEPROM. **Je však nutné správně nastavit zařízení před připojením do vozidla.** Tachografový vstup je aktivní po startu vždy a zpracovává data ihned po připojení napájecího napětí a vnitřní inicializaci chipu (do 100ms dle varianty firmware).

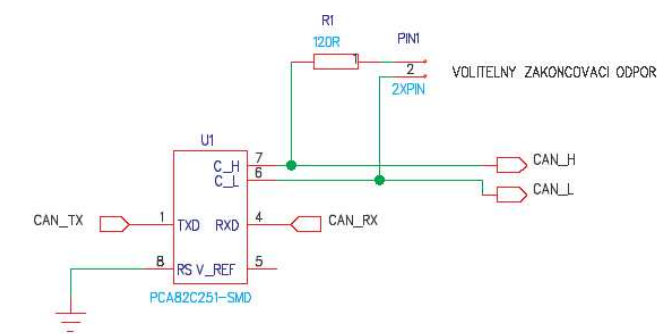
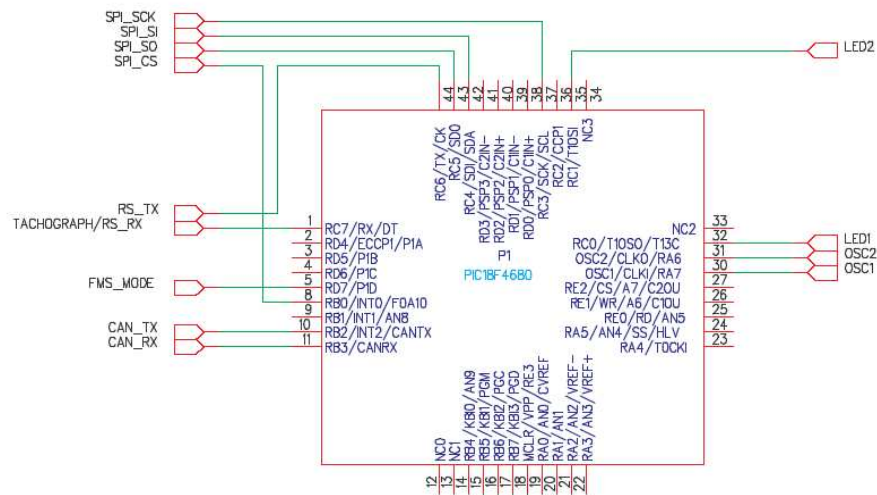
Chip poskytuje data ze sběrnice CAN která jsou dostupná odposlechem. Na různých vozidlech, modelových řadách a rocích výroby jsou dostupné různé kolekce zpráv.

Interface do nadřazeného systému

Pro připojení do nadřazeného systému (tedy toho, který využívá FMS OEM CHIP k získávání dat z vozidla) je možné využít jednoho ze dvou možných rozhraní a to SPI nebo RS232. Výběr použitého rozhraní se volí logickou úrovní, která je přivedena na pin RD7. Je li na tomto pinu po zapnutí napájení logická 0, je jako rozhraní použito RS232. V případě logické 1 pak využívá nadřazený systém rozhraní SPI.

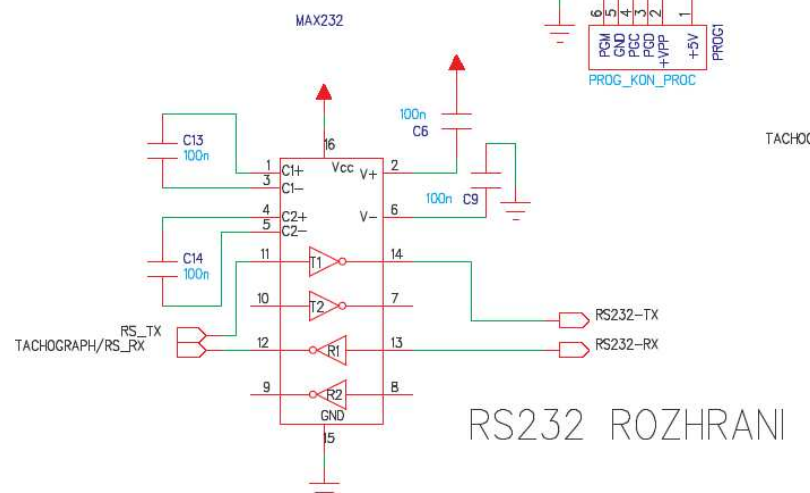
Režim RS232 dovoluje připojit pouze sběrnici CAN. Režim SPI dovoluje připojit sběrnici CAN a digitální tachograf.

SPI INTERFACE

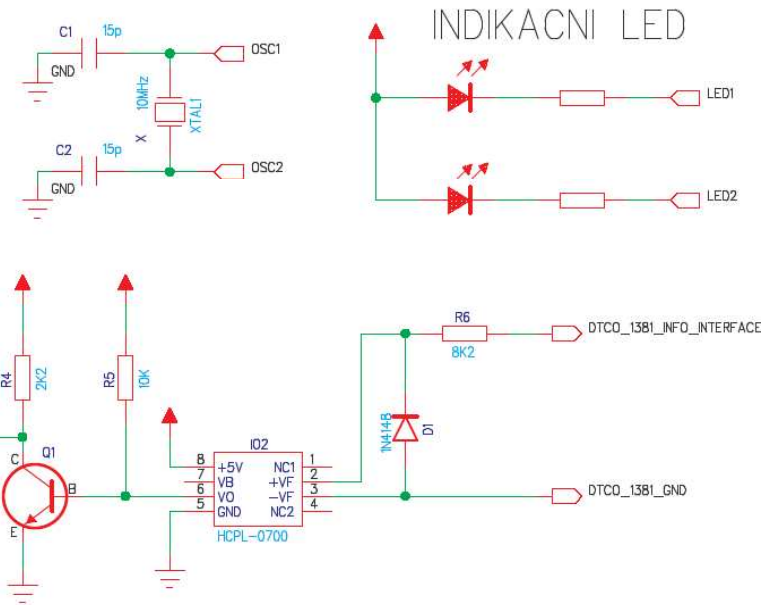


PRIKLAD ROZHRANI CAN

PROGRAMOVACI KONEKTOR



RS232 ROZHRANI



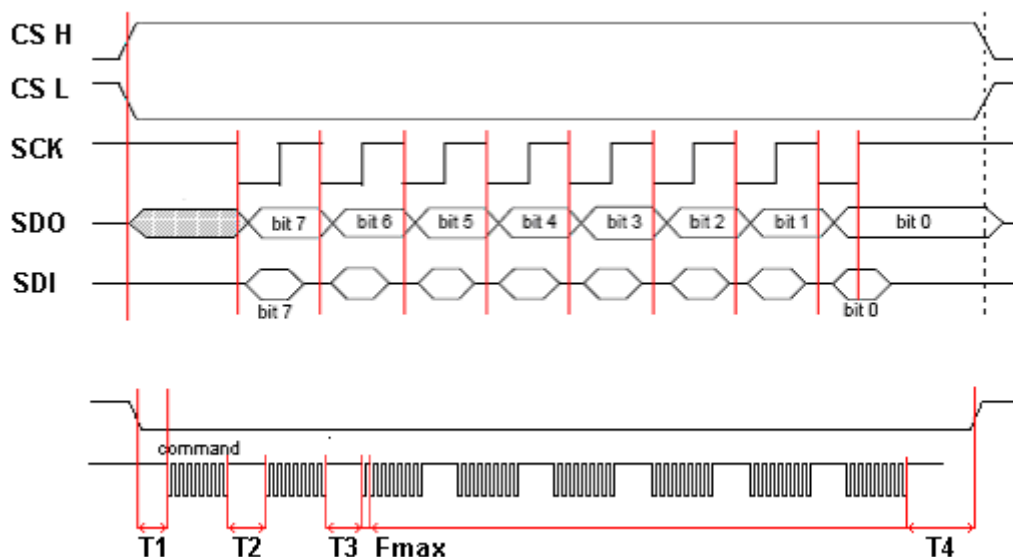
PRIKLAD PRIPOJENI K DIGITALNIMU TACHOGRAFU

Obr. 2: Typické zapojení obvodu.

SPI interface

FMS OEM CHIP poskytuje data prostřednictvím rozhraní SPI. Rozhraní pracuje v SPI módu 01. Obvod lze dodat i ve variantě s aktivní úrovní signálu CS ve stavu H. Maximální rychlost SPI (F_{SCKMAX}) je 2 MHz. Při komunikaci po SPI je nutno dodržet tato pravidla:

- $F_{SCK} \leq 3.00\text{MHz}$
- čas mezi přechodem signálu CS do aktivní úrovně (logická 0) a odesláním-čtením prvního bajtu ($T1$) musí být $\geq 12\mu\text{s}$.
- čas mezi ukončením přenosu bajtu command a počátkem čtení prvního datového bajtu přenosu ($T2$) $\geq 12\mu\text{s}$.
- čas mezi ukončením přenosu datového bajtu a čtením dalšího datového bajtu ($T3$) $\geq 1\mu\text{s}$.
- čas po přenosu posledního datového bajtu a deaktivací CS ($T4$) $\geq 10\mu\text{s}$.
- aby mohla být zaručena rychlost zápisu dat do interního registru SPI v režimu SPI slave, je v okamžiku aktivního signálu CS je pozastaveno zpracování dat z CAN sběrnice. Příjem dat do HW registrů integrovaného CAN řadiče je i nadále aktivní, tudíž v praxi nedochází k významné ztrátě dat. Důležitá data na CAN sběrnici se periodicky opakují a tedy maximálně dojde k jejich pozdější aktualizaci. Jeho délka je závislá na délce aktivace signálu CS.
- je doporučena mezi dvěma po sobě jdoucími čteními (deaktivace CS) při prvním čtením a aktivací CS při následním čtení vložit mezeru 1ms.



CS H – Varianta firmware s aktivním Signálem CHIP SELECT v logické 1.
 CS L – Varianta firmware s aktivním Signálem CHIP SELECT v logické 0.
 Variantu signálu CS si volí zákazník dle jeho potřeby.

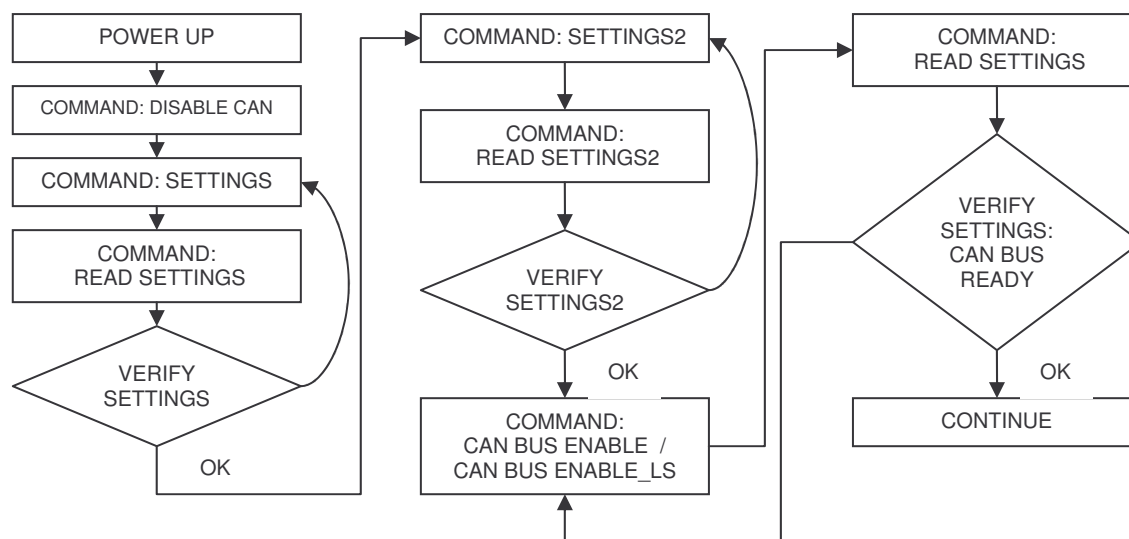
Obr. 3: Rozhraní SPI

Přenos na SPI zahajuje master zasláním bajtu „command“. Tento bajt specifikuje další činnost. Za tímto bajtem může následovat pole dat čtených z FMS OEM CHIPu. Jednotlivé příkazy jsou specifikovány v následující tabulce:

Název příkazu	Hodnota příkazového bajtu (binárně) b7 b0	Popis
READ	0000 0XXX Následuje některá z datových struktur popsaných dále.	XXX: 000 – Verze firmware 001 – CAN bus data 010 – Tachograph data 110 – Settings 2 111 – Settings
SETTINGS	01ZY YXXX 0011 0011 (51d) 1010 0101 (165d)	Z: 0 – Normální mód CANu 1 – Listen only mód CANu YY: 00 – Standardní 11 bitový CAN ID 01 – Rozšířený 29 bitový CAN ID 11 – Standardní i rozšířený 29 bitový CAN ID XXX : Rychlost CAN sběrnice 0001 – 62.5k 0010 – 83.3k 0011 – 100k 0100 – 125k 0101 – 250k 0110 – 500k 0111 – 1M
SETTINGS2	1XXX XXXX (high byte) XXXX XXXX (low byte) RRRR RYYY 1010 0101 (165d) 0011 0011 (51d)	X-přídavná informace pro výběr typu vozidla (CAR TYPE). Tato informace je vyžadována v případech, kdy u 2 různých typů vozidel nese CAN zpráva se stejným identifikátorem stejnou informaci. Y-typ tachografu,0-VDO Siemens, 1-Stoneridge,2-Actia R-rezerva
CAN BUS ENABLE	0001 0001 (17d) 0011 1100 (60d) 0101 1010 (90d)	Povoluje CAN sběrnici.Listen only podle nastavení settings.
CAN BUS ENABLE LS	0001 0010 (18d) 0111 1110 (126d)	Povoluje CAN sběrnici „natvrdo“ jen v listen only modu.

	1010 0101 (165d)	
CAN BUS DISABLE	0001 0000 (16d) 1010 0101 (165d) 0000 1111 (15d)	Zakazuje CAN sběrnici.
SEND OBD DTC REQ.	0001 1000 (24d) 0000 0XXX	Zaslání OBD dotazu. Kód OBD DTC dotazu. 1- 11bitové ID, mode 3 2- 11bitové ID, mode 7 3- 29bitové ID, mode 3 4- 29bitové ID, mode 7
READ OBD DTC TABLE	0001 1111 (31d) Následuje datová struktura popsaná dále.	Čtení OBD DTC kódů v paměti.
SET CAN USR RCV MSG	0001 0011 (19d) Následuje datová struktura popsaná dále.	Příkaz k nastavení uživatelského bufferu pro příjem zpráv z CANu.
DISABLE ALL CAN USR RCV MSG	0001 0100 (20d)	Příkaz zakáže všechny uživatelské buffery pro příjem dat z CANu.
SEND CAN USR MSG	0001 0110 (22d) Následuje datová struktura popsaná dále.	Příkaz pro uživatelské odeslání dat na CAN.
READ CAN USR RCV MSG	0001 0111 (23) 0000 0AAA Následuje datová struktura popsaná dále.	Příkaz pro uživatelské čtení dat z CANu – čtení uživatelského bufferu. AAA-index bufferu 1..7

Příkaz **SETTINGS** na **SPI**



Příkaz **READ** na **SPI**

Příkaz READ je určen k blokovému čtení dat z FMS OEM CHIPu. Master odešle první datový bajt s příkazem READ, a přijme hodnotu 0. Dalším čtením z SPI přijímá data. Čtení je ukončeno po přečtení celé velikosti struktury dat, může však být ukončeno i dříve vrácením signálu CS do neaktivní úrovně (logická 1).

READ: FIRMWARE

Tento příkaz je určen ke čtení verze firmware FMS OEM CHIPu. Délka čtených dat je 13 bajtů. Celý přenos je tedy dlouhý 14 bajtů, 1 bajt pro příkaz READ FIRMWARE a 13 datových bajtů. Vrácená hodnota je textový řetězec s označením verze firmware, například „FMSOEMV5002CT__“. Řetězec není ukončen žádným zakončovacím znakem.

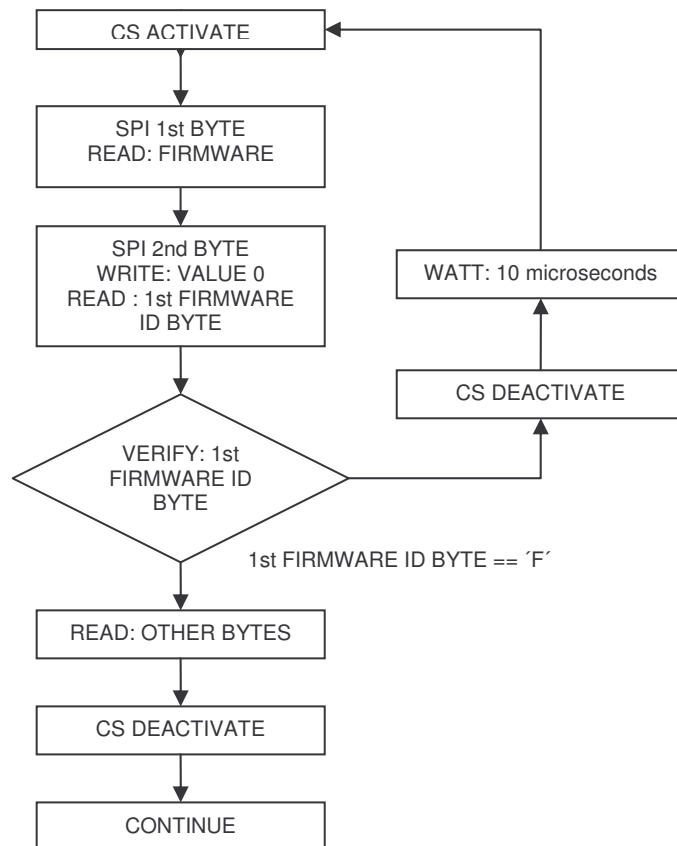
FMSOEMV5002CT__ (FMSOEMV5yyyab__)

yyy - verze firmware

a - C-podporuje CAN, X-nepodporuje CAN

b - T-podporuje digitální tachografy, X-nepodporuje tachografy

__ - rezerva (2 znaky)



READ: CAN BUS DATA

Tento příkaz je určen ke čtení dat z rozhraní CAN. Jsou čtena data, která tvoří následující strukturu:

```

#define BYTE      unsigned char      // 8 bitový datový typ bez znaménka
#define UINT      unsigned __int16   // 16 bitový datový typ bez znaménka
#define SINT      signed __int16     // 16 bitový datový typ se znaménkem
#define ULONG     unsigned __int32   // 32 bitový datový typ bez znaménka
  
```

Více bytové typy jsou ve formátu little-endian.

Definice struktury FMS do verze 1.10 včetně:

```

typedef struct _FMS {
    BYTE  begin_check;           // sizeof(FMS) = 0x52
    UINT  rpm;
    UINT  speed;
    BYTE  acc_pedal;
    BYTE  brake_pedal;
    ULONG total_fuel_used;
    ULONG total_engine_hours;
    BYTE  fuel_level;
    UINT  fuel_consumption;
    BYTE  axle_weight_captured[12];
    BYTE  axle_weight_location[12];
    UINT  axle_weight[12];
    ULONG total_vehicle_distance;
    UINT  daily_vehicle_distance;
}
  
```

```

        SINT  service_distance;
        BYTE  engine_coolant_temperature;
        UINT  tachograph_speed;
        BYTE  tachograph[4];
        BYTE  end_check;           // 0xAA
    } FMS;

```

Definice struktury FMS od verze 1.11 včetně:

```

typedef struct _FMS {
    BYTE  begin_check;           // sizeof(FMS) = 0x7C
    UINT  rpm;
    UINT  speed;
    BYTE  acc_pedal;
    BYTE  brake_pedal;
    ULONG total_fuel_used;
    ULONG total_engine_hours;
    BYTE  fuel_level;
    UINT  fuel_consumption;
    BYTE  axle_weight_captured[12];
    BYTE  axle_weight_location[12];
    UINT  axle_weight[12];
    ULONG total_vehicle_distance;
    UINT  daily_vehicle_distance;
    SINT  service_distance;
    BYTE  engine_coolant_temperature;
    UINT  tachograph_speed;
    BYTE  tachograph[4];
    BYTE  tire_pressure_captured[12];
    BYTE  tire_pressure_location[12];
    BYTE  tire_pressure[12];
    UINT  door;
    UINT  fuel_instantaneous;
    UINT  fuel_rate;
    BYTE  end_check;
} FMS;

```

Platná hodnota položek `begin_check` a `end_check` označuje platná data. Pokud tyto 2 položky nemají správné hodnoty, je třeba data z vozidla ignorovat. Jsou-li všechny byty položek nastaveny na `0xFF`, je indikováno nepřijetí těchto dat z CANu.

Data ze struktury je možné převést na skutečné hodnoty pomocí této tabulky:

Data	Počet bitů	Váha 1 bitu	Offset
Rychlost	16	1/256 km/h	0
Poloha pedálu akcelerace	8	0.4 %	0
Poloha brzdového pedálu	8	0.4 %	0
Celkově spotřebované palivo	32	0.5 litru	0
Stav palivové nádrže	8	0.4 % SAE1939 nebo litry v nádrži	0
Otáčky motoru	16	0.125 otáčky	0
Zatížení nápravy	16	0,5 kg	0
Celkový počet motohodin	32	0,05 hod.	0
Celkově najeté kilometry	32	5 m	0
Vzdálenost v kilometrech do servisní prohlídky	16	5 km	-160 635
Teplota chladící kapaliny.	8	1 °C	-40

Průměrná spotřeba.	16	1/512 km/L – SAE1939 L/100km - VW	0
--------------------	----	---	---

Axle weight

Položka `axle_weight_location[x]` udává lokaci hodnoty zatížení nápravy v položce `axle_weight[x]`. Hodnota `axle_weight_captured[x] = 0` udává, že položka neobsahuje žádnou (platnou) hodnotu, `axle_weight_captured[x] = 1` udává, že položka obsahuje platnou hodnotu.

V položce `axle_weight_location[x]` je zakódována informace o čísle měřené nápravy a kole této nápravy. Dolní 4 bity udávají index kola, horní 4 bity udávají index nápravy. Jsou-li všechny 4 bity nastaveny na 1, lokace není známa.

Položka tachograph[4]

Tato položka obsahuje informace, které je možno dekodovat dle následujícího popisu:

tachograph[0]

Bit 2..0 :Driver 1 working state

- 000 = Rest
- 001 = Driver available
- 010 = Work
- 011 = Drive
- 110 = Error
- 111 =

Bit 5..3 :Driver 2 working state

- 000 = Rest
- 001 = Driver available
- 010 = Work
- 011 = Drive
- 110 = Error
- 111 = not available

Bit 7..6 :Drive recognize

- 00 = Vehicle motion not detected
- 01 = vehicle motion

tachograph[1]

Bit 3..0 : Driver 1 time rel states

- 0000 = normal
- 0001 = 15 min bef. 4 ½ h
- 0010 = 4 ½ h reached
- 0011 = 15 min bef. 9 h
- 0100 = 9 h reached
- 0101 = 15 min bef. 16 h
- 0110 = 16h reached
- 1110 = Error
- 1111 = not available

Bit 5..4 :Driver 1 card

00 = Card not present
01= Card present

Bit 7..6 :Overspeed

00 = No overspeed
01 = Overspeed

tachograph[2]**Bit 3..0 : Driver 2 time rel states**

0000 = normal
0001 = 15 min bef. 4 ½ h
0010 = 4 ½ h reached
0011 = 15 min bef. 9 h
0100 = 9 h reached
0101 = 15 min bef. 16 h
0110 = 16h reached
1110 = Error
1111 = not available

Bit 5..4 :Driver 2 card

00 = Card not present
01= Card present

Bit 7..6 :Not used**tachograph[3]****Bit 0..1 :System event**

00 = no tachogr. Event
01 = tachogr. Event

Bit 2..3 :Handling information

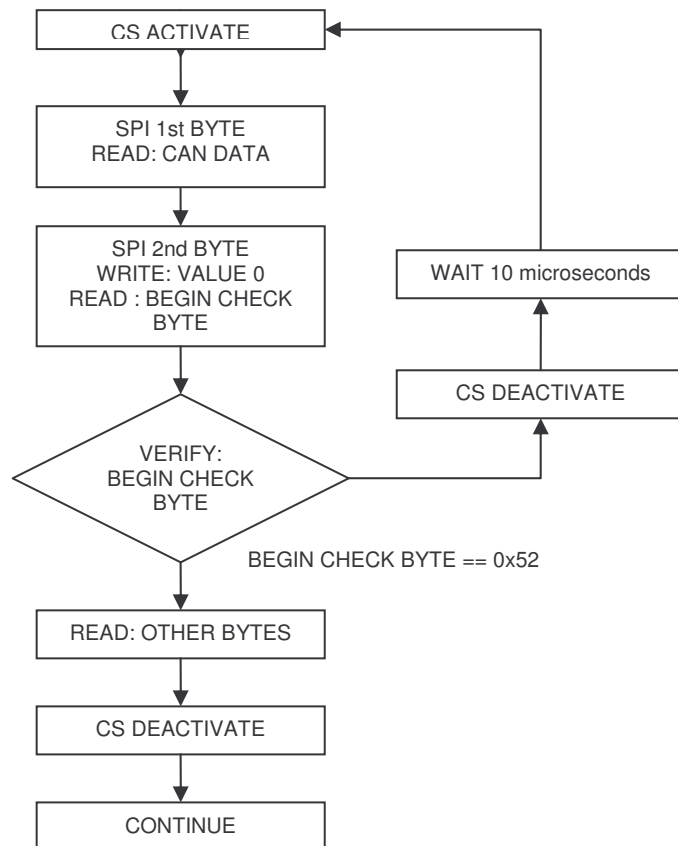
00 = no handling information
01 = handling information

Bit 5..4 :Tachgraph performance

00 = Normal performance
01 = Performance

Bit 7..6 :Direction indicator

00 = Forward
01 = Reverse

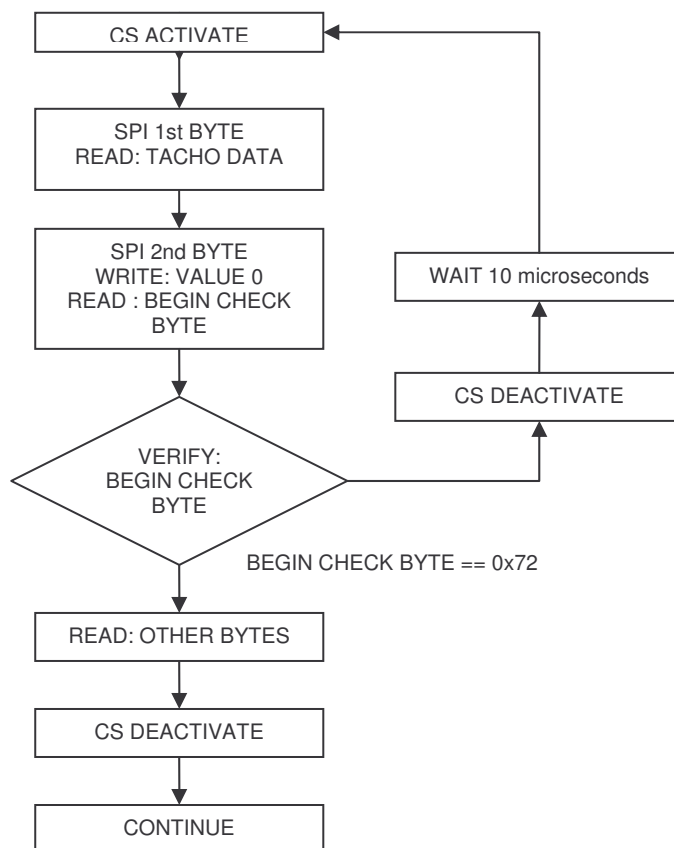


READ: TACHOGRAPH DATA

```

typedef struct _DTI{
    BYTE    begin_check;           // sizeof(DTI) = 0x72
    BYTE    seconds;
    BYTE    minutes;
    BYTE    hours;
    BYTE    month;
    BYTE    day;
    BYTE    year;
    BYTE    local_minute_offset;
    BYTE    local_hour_offset;
    BYTE    work_states;
    BYTE    driver_1_states;
    BYTE    driver_2_states;
    BYTE    tachograph_status;
    UINT    tachograph_vehicle_speed;
    LONG    total_vehicle_distance;
    LONG    trip_distance;
    INT     k_factor;
    INT     engine_speed;
    INT     additional_information;
    BYTE    vehicle_id_len;
    BYTE    vehicle_id[20];
    BYTE    vehicle_reg_len;
    BYTE    vehicle_reg[20];
    BYTE    driver_1_len;
    BYTE    driver_1[20];
    BYTE    driver_2_len;
    BYTE    driver_2[20];
    BYTE    end_check;           // 0xAA
}
  
```

```
} DTI;
```



READ: SETTINGS

Tento příkaz je určen ke zpětnému čtení aktuálního nastavení. Odpovědí je jeden datový bajt, jehož bity mají následující význam:

AAZY YXXX

AA: 00 – CAN bus OFF
01 – CAN bus ON
10 – Startup CAN

Z: 0 – Normální mód CANu
1 – Listen only mód CANu

YY: 00 – Standardní 11 bitový CAN ID
01 – Rozšířený 29 bitový CAN ID
10 – Standardní i rozšířený 29 bitový CAN ID

XXX : Rychlost CAN sběrnice
0001 – 62.5k
0010 – 83.3k
0011 – 100k
0100 – 125k
0101 – 250k

0110 – 500k
0111 – 1M

READ: SETTINGS2

Po prvním datovém bajtu s příkazem čtení SETTINGS2 následují dva datové bajty s typem vozidla a třetí bajt s nastaveným typem tachografu.

Vozidlo	Car type	CAN
FMS/SAE1939	0	250k, ext
VW	0	500k, st
OBD 11 bitový	1024	250/500k, st
OBD 29 bitový	1025	250/500k, ext
Mercedes Sprinter	48	500k, st
Mercedes Vito	49	500k, st
Nisan/Renault	64	250/500k, st
Nisan/Renault	65	250/500k, st
Toyota	96	500k, st
Ford Mondeo	128	500k, st
Ford Tranzit	129	500k, st
Ford CMAX	130	500k, st
Fiat	144	250/500k, st
Mazda	160	500k, st
Suzuki SX4	176	500k, st

READ OBD DTC TABLE

```
typedef struct _OBD_DTC_TABLE{
    BYTE          size;
    INT           dtc[16];
} OBD_DTC_TABLE;
```

Za příkazem pro čtení tabulky chybových kódů následuje uvedená struktura. První bajt obsahuje počet platných DTC kódů v tabulce. Následuje maximálně 16 DTC kódů.

Tabulka je vymazána vždy při odeslání příkazu SEND OBD DTC REQ.

DTC kód je třeba interpretovat takto:

b7.....b0 b7.....b0
UUUUVVVV CCRRSSSS
(high byte) (low byte)

CC: 0: znak P RR: 0: znak 0
 1: znak C 1: znak 1
 2: znak B 2: znak 2
 3: znak U 3: znak 3

SSSS, UUUU, VVVV, hodnota 0..9 odpovídá znakům 0..9.

Například hodnota 0x73C0 je dekódována jako u0033. 0x2004 jako p0420.

SEND CAN USR MSG

```
typedef struct _SPI_USR_SEND_CAN_MESSAGE
{
    LONG id;
    BYTE data[8];
    BYTE len;
    BYTE flags;
    BYTE check[2];
}SPI_USR_SEND_CAN_MESSAGE;
```

len - počet datových bajtů 0..8
 flags - rozšířený identifikátor 0b00100000
 rtr frame 0b01000000
 check - check[0]= 0x3F (63d), check[1]= 0xC0 (192d)

SET CAN USR RCV MSG

```
typedef struct _RCV_MSG_TABLE_ITEM
{
    BYTE index;
    LONG id;
    BYTE flags;
    BYTE data[4];
    BYTE check;
}RCV_MSG_TABLE_ITEM;
```

index - index do tabulky uživatelsky přijímaných CAN zpráv
 id - identifikátor zprávy která je přijímána do buffer
 flags - b0-standardní (0)nebo rozšířený identifikátor
 b1-datový (0) nebo rtr(1) frame
 b2-je li 1 je filtrováno i podle datového bajtu 0
 b3-je li 1 je filtrováno i podle datového bajtu 1
 b4-je li 1 je filtrováno i podle datového bajtu 2
 b5-je li 1 je filtrováno i podle datového bajtu 3
 b6-příjem zprávy povolen
 data - datové bajty podle kterých je možné také filtrovat přijímané zprávy
 check - kontrolní bajt, hodnota 0x0F (15d)

READ CAN USR RCV MSG

```
typedef struct _READ_CAN_MESSAGE{
    LONG id;
    BYTE data[8];
    BYTE len;
    BYTE flags;
} READ_CAN_MESSAGE;
```


flags - rozšířený identifikátor 0b00100000
rtr frame 0b01000000

Rozdíly v SPI komunikaci proti FMS OEM CHIPu V4

- 1) Příkaz SETTINGS dovoluje nastavovan menší počet rychlostí CANu. Nevyužité rychlosti bez perspektivy použití byly vyřazeny.
- 2) Bit ušetřený snížením počtu rychlostí je využit pro možnost současného příjmu zpráv se standardním i rozšířeným identifikátorem.
- 3) Za příkazem SETTINGS následuje dvojice bajtů daných hodnot. Pokud nejsou zaslány je příkaz ignorován.
- 4) Příkaz SETTINGS po startu neaktivuje CAN. Je k tomu třeba použít samostatný příkaz CAN BUS ENABLE nebo CAN BUS ENABLE LS.
- 5) Příkaz SETTINGS2 je rozšířen na 15 bitů. Dále následuje bajt s nastavením tzpu tachografu a 2 kontrolní bajty podobně jako u příkazu SETTINGS.
- 6) Doplněna možnost číst SETTINGS 2.

RS232 interface

Současná verze firmware generuje tato data:

Znak	ASCII - dec	ASCII – hex	Popis
R	82	52	Otáčky motoru
S	83	53	Rychlost vozidla
A	65	41	Poloha pedálu akcelerace.
B	66	42	Poloha brzdového pedálu.
F	70	46	Hodnota celkově spotřebovaného paliva.
H	72	48	Celkový počet motohodin.
L	76	4C	Stav palivové nádrže. V kamionu procenta, VW-litry
C	67	43	Průměrná spotřeba.
W *	87	57	Zatížení nápravy.
T	84	54	Celkově najeté kilometry.
D	68	44	Denní nájezd kilometrů.
V	86	56	Vzdálenost v kilometrech do servisní prohlídky.
N	78	4E	Teplota chladicí kapaliny.
I	73	49	Jméno řidiče z tachografové karty
K	75	4B	Překročení délky jízdy bez bezpečnostní přestávky dle tachografu.
E	69	45	Chybové kódy
Z	90	5A	Tachografová rychlost
J	74	4A	Průtok paliva
P	80	50	Aktuální spotřeba.
U*	85	55	Tlak v pneumatikách
Q	81	51	Bitové stavy dveře, kufr apod. 16 bitové číslo uvedené hexadecimálně. Bit 0 - zavřeno.

*Příkaz v odpovědi přenáší zatížení (nebo tlak v pneumatikách) na více nápravách. Před každou hodnotou zatížení nápravy jsou uvedeny dva znaky, které specifikují lokaci změřeného zatížení. První znak specifikuje číslo nápravy, druhý pak pozici kola. Indexy nápravy mají rozsah 0-15 a jsou zadávány znaky 0...F, tzn hexadecimálně. Hodnota F znamená že poloha není známa.

Teoreticky tak zpráva se zatížením náprav může obsahovat zatížení až 15 náprav takovouto sekvencí:

W15:0-F-3245:1-F-2252:0-F-1763.....E-F-1223

Zatížení na první nápravě (přední) je 3245 kg, lokace kola ne kterém je měřeno není známa (F), zatížení další nápravy je 2252kg, a tak dále.

Zařízení je schopno automaticky generovat přednastavená data s nastavitelnou periodou generování 1s - 255 minut.

Konfigurace je po startu načtena z interní EEPROM. Do EEPROM je nastavení uložen při každé změně nastavení

Rozsahy jednotlivých veličin jsou dány takto:

Data	Počet číselných znaků	Váha 1 bitu
Rychlost	1..3	km/h
Poloha pedálu akcelerace	1..3	%
Poloha brzdového pedálu	1..3	%
Celkově spotřebované palivo	1..10	litry
Stav palivové nádrže	1..3	%-nakladní /litry- osobní
Otáčky motoru	1..4	rpm
Zatížení nápravy	1..5	kg
Celkový počet motohodin	1..10	hod.
Celkově najeté kilometry	1..8	Km
Vzdálenost v kilometrech do servisní prohlídky	1..8	Km
Teplota chladící kapaliny.	1..3	°C
Průměrná spotřeba.	1..5	0.1litrů/100 km
Okamžitá spotřeba.	1..5	0.1litrů/100 km
Průtok paliva.	1..5	0.05litrů/hod
Tlak v pneumatikách	1.4	kPa

Příklady komunikace:

Komunikační řetězec obsahuje kontrolní kód. Tento kód je xor znaků které následují za znakem \$ do znaku * Tyto znaky nejsou v kontrolním kódu zahrnuty.

Start

\$PCAN,C,VER,1.10,CANLABsro,*41<enter>

Po startu zasílá zařízení na RS232 1x uvedenou informaci. Text 1.10 identifikuje verzi firmware a v každé nové verzi je hodnota změněna. Jako znak <enter> je použita dvojice znaků CR LF; 0X0D 0x0A (konvence Win/DOS).

Čtení verze firmware

```
$PCAN,C,FW,*62<enter>
```

Odpověď

```
$PCAN,C,VER,1.10,FMS_CHIP_V5,CANLABsro,*44<enter>
```

Dotaz na zjištění verze firmware. Odpověď má stejný formát jako paket Start.

Žádost o data zasláná do CAN jednotky z nadřazeného systému

```
$PCAN,C,GET,#Q##,*7B<enter>
```

Požaduje se údaj o stavu dveří a krytu.

```
$PCAN,C,GET,Q0000,*58<enter>
```

Žádné dveře či kryt nejsou otevřeny. 4 znaky za symbolem Q jsou hexadecimální číslo s kdy jednotlivé bity reprezentují stav dveří.

Pro nákladní vozidla platí že je indikován stav

n stav:

Q0000 - všechny dveře zavřeny

Q0001 - některé dveře otevřeny

Pro vozidla VW:

Q0000 - všechny dveře zavřeny

Q0001 - otevřeny dveře řidiče

Q0002 - otevřeny dveře spolujezdce

Q0004 - otevřeny levé zadní dveře

Q0008 - otevřeny pravé zadní dveře

Q000F - otevřeny všechny 4 dveře

Q0010 - otevřena kapota motorového prostoru

Q0020 - otevřena kapota zavazadlového prostoru

Žádost o data zasláná do CAN jednotky z nadřazeného systému

```
$PCAN,C,GET,#W##,*7D<enter>
```

Požaduje se údaj o zatížení náprav – znak W viz. tabulka.

Odpověď z CAN jednotky

```
$PCAN,C,GET,W3:1-F-5384:2-F-0:0-F-5343,*19<enter>
```

Jednotka signalizuje, že bylo změřeno zatížení na 3 nápravách, vrácená data signalizují lokaci nápravy a hodnotu zatížení. Náprava č. 1, 2 a 0, údaj o kole na kterém se měří není k dispozici (F) a hodnota v kg. Jako oddělovač mezi nápravami je použit znak ':' (dvojtečka).

Alternativní odpověď z CAN jednotky

```
$PCAN,C,GET,WX,*06<enter>
```

Jednotka signalizuje, že požadovaná data nejsou k dispozici (za znakem W leží znak X který označuje data která nejsou změřena - nebyla přijata z CAN sběrnice).

Žádost o data zaslaná do CAN jednotky z nadřazeného systému

```
$PCAN,C,GET,#SRT##,*7F<enter>
```

Je požadován údaj o rychlosti, otáčkách motoru a celkovém nájezdu kilometrů (znaky S, R a T).

Odpověď z CAN jednotky

```
$PCAN,C,GET,S16R1313T398405,*58<enter>
```

Jednotka vrací údaj o rychlosti (16km/h), otáčkách (1313 rpm/min) a stav tachometru (398405 km).

Alternativní odpověď z CAN jednotky

```
$PCAN,C,GET,SXRXTX,*04<enter>
```

Jednotka vrací údaj o tom, že data nejsou k dispozici. Pravděpodobnost neexistence těchto dat je nízká, proto u takovéto zprávy se dá předpokládat ze došlo k přerušení CANovského vedení k jednotce, závadě/špatnému nastavení CANu nebo vozidla na jednotce, nebo je jednotka aktivní, avšak vozidlo je vypnuté (CAN vozidla je vypnut) – chyba signálu 15.

Periodicky generovaná data podle přednastavené masky mají stejný tvar jako odpovědi jednotky.

Žádost o data zaslaná do CAN jednotky z nadřazeného systému

```
$PCAN,C,GET,#K##,*61<enter>
```

Požaduje se údaj o překročení délky jízdy bez přestávky pokud je tato informace z digitálního tachografu přenášena na CAN sběrnici.

```
$PCAN,C,GET,K77,*42<enter>
```

Za znakem leží 2 hexadecimální kódy, každý udává stav pro jednu z karet v tachografu. Kódy mají tento význam:

- 0 = Rest
- 1 = Driver available
- 2 = Work
- 3 = Drive
- 6 = Error
- 7 = Not available

Zpráva s nastavením do CAN jednotky 1

`$PCAN,C,SET,C5,EXT,LISO,P10,T0,#SRTL##,*18<enter>`

- C5** - nastavení CANovské rychlosti (obvykle 5-kamiony, 6-motorovy CAN u VW)
- EXT/TWO** - typ CANovského identifikátoru, EXT označuje rozšířený identifikátor, není-li EXT uvedeno, je nastaven standardní identifikátor (rozšířený identifikátor – kamiony, neuveden-standardní u VW)
Je-li uvedeno TWO jsou přijímány oba typy identifikátorů
- P10** - generovat automaticky data každých 10 minut, není-li identifikátor P uveden, data nejsou automaticky generována.
- LISO** - je-li identifikátor uveden, zařízení pracuje v módu, který zabezpečuje, že jednotka nemůže ovlivnit práci CAN sběrnice ve vozidle
- T<n>** - typ vozidla, použito v případech, kdy 2 stejné identifikátory CAN zpravnesou u různých vozidel různá data, hodnoty: 0-kamiony ; 16-Basic VW ; 48-Basic Mercedes ; 64- Basic Nisan ; 80- Basic Opel ; 96- Basic Toyota
- #SRTL##** - mezi znaky # a ## je pak seznam automaticky generovaných dat. V tomto případě S-rychlost, R-otáčky, T-stav tachometru, L-stav nádrže.

Po obdržení této zprávy jednotka data zpracuje, uloží (EEPROM), odpoví zpět zasláním stejné zprávy a provede svůj restart. Po restartu pokračuje v činnosti dle nového nastavení.

`$PCAN,C,SET,C5,EXT,LISO,P0+10,T0,#SRABFHLCTDVNIWU##,*5A<enter>`

A každých 10 sekund je generována zpráva

`$PCAN,C,GET,S45R1432A88B0F2428H341L56CXT33457DXVXN93IX,*7B <enter>`

Protože je požadováno i zatížení náprav a tlak v pneumatikách, kdy tyto 2 zprávy mohou nést každá až 12 údajů, jsou tyto 2 zprávy generovány samostatně pro zjednodušení jejich parsování. Jsou tedy celkově vygenerovány 3 zprávy:

`$PCAN,C,GET,S45R1432A88B0F2428H341L56CXT33457DXVXN93IX,*7B <enter>`

`$PCAN,C,GET,W2:0-F-5792:1-F-2055,*66<enter>`

`$PCAN,C,GET,UX,*04<enter>`

Zpráva s nastavením do CAN jednotky 2

```
$PCAN,C,SET,C5,EXT,LISO,P1+40,T0,#SRTL##,*07<enter>
```

Zpráva je stejná jako předchozí. Jediným rozdílem je perioda generování. Ta je nastavena na 1 minutu a 40 sekund, tedy celkově 100 sekund.

Zpráva s nastavením do CAN jednotky 3

```
$PCAN,C,SET,C5,EXT,LISO,P0,T0,*3F<enter>
```

Zpráva je stejná jako předchozí co do nastavení CANu. Nejsou však generovány periodické zprávy. Zařízení tedy poskytuje data na dotaz.

Uživatelské zaslání dat na CAN

```
$PCAN,C,CAN,S,I123,S,L5,B0:11,B1:22,*3E<enter>
```

Zpráva z nadřazeného systému do FMS CHIPu. Po přijetí této zprávy odešle CHIP na CAN zprávu se standardním ID 123 s 5 datovými bajty. Hodnoty datových bajtů budou 11,22,0,0,0.

Uživatelské čtení dat z CANu

```
$PCAN,C,CAN,T0,I123,S,B0:11,B1:22,*5C<enter>
```

Zpráva z nadřazeného systému nastaví uživatelský buffer 0 pro čtení CANu na příjem zprávy se standardním ID 123. Aby byla zpráva přijata musí kromě stejného identifikátoru obsahovat v datovém bajtu 0 hodnotu 11 a v datovém bajtu 1 hodnotu 22. Platí že pokud je datový bajt uveden je podle jeho hodnoty také filtrováno. Identifikátor musí být uveden vždy. Filtraci lze provádět podle identifikátoru a volitelně i podle prvních 4 datových bajtů. K příjmu je k dispozici 8 uživatelských bufferů T0 .. T7.

```
$PCAN,C,CAN,T1,I1234,E,*7E<enter>
```

Zpráva nastaví uživatelský buffer 1 pro čtení CANu na příjem zprávy s rozšířeným identifikátorem. Filtrace příjmu do bufferu je prováděna jen na základě identifikátoru.

```
$PCAN,C,CAN,R0,*71<enter>
```

Dotaz z nadřazeného na hodnotu uživatelského bufferu 0.

```
$PCAN,C,CAN,R0,E,*18<enter>
```

Chip odpovídá na předchozí dotaz na hodnotu uživatelského bufferu 0 hodnotou E – empty. Zpráva na kterou buffer reaguje buď ještě nebyla z CANu přijata nebo již byla

přečtena. Každé čtení nastaví buffer na stav empty do doby než je zpráva znovu přijata z CANu.

```
$PCAN,C,CAN,R0,I123,S,L8,B0:11,B1:22,B2:3,B3:4,B4:5,B5:6,B6:7,B7:8,*08<enter>
```

Chip odpovídá na předchozí dotaz na hodnotu uživatelského bufferu 0. Buffer obsahuje zprávu se standardním ID 123 o délce 8 datových bajtů s hodnotami 11,22,3,4,5,6,7,8.

```
$PCAN,C,CAN,R1,*70
```

Dotaz z nadřazeného na hodnotu uživatelského bufferu 0.

```
$PCAN,C,CAN,R1,I1234,E,L8,B0:11,B1:22,B2:3,B3:4,B4:5,B5:6,B6:7,B7:8,*2B
```

Chip odpovídá na předchozí dotaz na hodnotu uživatelského bufferu 1. Buffer obsahuje zprávu s rozšířeným ID 123 o délce 8 datových bajtů s hodnotami 11,22,3,4,5,6,7,8.

```
$PCAN,C,CAN,D,*57
```

Zpráva vymaže všechny uživatelské buffery.

Příkaz SETTINGS u nejčastěji monitorovaných vozidel.

Nákladní vozidla – páteřní CAN bus

- listen only
- rozšířený CAN ID
- rychlost 250k

```
$PCAN,C,SET,C5,EXT,LISO,P0+10,T0,#SRTL##,*03<enter>
```

Nákladní vozidla – FMS gateway (FMS brána)

- normální mód
- rozšířený CAN ID
- rychlost 250k

```
$PCAN,C,SET,C5,EXT,P0+10,T0,#SRTL##,*36<enter>
```

Vozidla Škoda/VW, motorový CAN bus

- listen only
- standardní CAN ID
- rychlost 500k

```
$PCAN,C,SET,C6,LISO,P0+10,T16,#SRTL##,*52<enter>
```

U vozidel VW se FMS OEM CHIP připojuje na motorový CAN. Nelze tak provádět čtení DTC kódů.

Vozidla Nissan, Renault, Dacia varianta 1 (Nissan Micra)

- listen only
 - standardní CAN ID
 - rychlost 500k
 - experimentální podpora
- ```
$PCAN,C,SET,C6,LISO,P0+10,T64,#SRTL##,*57<enter>
```

**Vozidla Nissan, Renault, Dacia varianta 2**

- listen only
  - standardní CAN ID
  - rychlost 250k
  - experimentální podpora
- ```
$PCAN,C,SET,C5,LISO,P0+10,T65,#SRTL##,*55<enter>
```

Vozidla Ford Mondeo

- listen only
 - standardní CAN ID
 - rychlost 500k
 - experimentální podpora
- ```
$PCAN,C,SET,C6,LISO,P0+10,T128,#SRTL##,*6E<enter>
```

**Vozidla Ford Tranzit**

- listen only
  - standardní CAN ID
  - rychlost 500k
  - experimentální podpora
- ```
$PCAN,C,SET,C6,LISO,P0+10,T129,#SRTL##,*6F<enter>
```

U dalších vozidel které podporují CAN diagnostiku lze číst některá obecná data:

- teplota motoru
- otáčky motoru
- rychlost vozidla
- pedál akcelerace
- stav paliva

Funkce je v experimentální stádiu.

Obecné OBD vozidlo, 11 bitový ID, 250k

```
$PCAN,C,SET,C5,P0+10,T1024,#SRTL,*69<enter>
```

Obecné OBD vozidlo, 11 bitový ID, 500k

```
$PCAN,C,SET,C6,P0+10,T1024,#SRTL##,*68<enter>
```

Obecné OBD vozidlo, 29 bitový ID, 250k

```
$PCAN,C,SET,C5,EXT,P0+10,T1025,#SRTL##,*0D<enter>
```

Obecné OBD vozidlo, 29 bitový ID, 500k

```
$PCAN,C,SET,C6,EXT,P0+10,T1025,#SRTL##,*2C<enter>
```


Čtení chybových kódů vozidla

OBD kompatibilní vozidla dovolují číst chybové kódy (DTC). Tyto kódy popisují chyby ve vozidle detekované řídicími jednotkami. CAN musí být připojen na diagnostický konektor a nesmí být aktivován mód LISTEN ONLY. Je-li tento mód ve vozidle aktivní, je možné jej deaktivovat (přednastavit nastavení s vypnutým listen only modem, přečíst chyby vozidla a opět jej aktivovat). Vozidlo musí podporovat OBD diagnostiku přes rozhraní CAN.

Zda OBD diagnostika v této standardizované formě funguje závisí na typu vozidla. Nicméně zaslání dotazu do vozidla, které tuto funkci nepodporuje nemá na vozidlo žádný vliv.

Za dotazem na DTC chyby jsou uváděny 2 parametry:

s – standardní CAN identifikátor
e – rozšířený CAN identifikátor

Jaký identifikátor je použit závisí na typu vozidla, nejjednodušší je otestovat nejprve standardní, pokud nefunguje zkusit méně častý rozšířený.

3 - diagnostika mód 3, uložené chybové kódy
7 - diagnostika mód 7, chybové kódy v současné nebo minulé jízdě

Funkce je v experimentální stádiu.

Dotaz 1:

```
$PCAN,C,GET,#E<s3>##,*2D<enter>
```

Odpověď:

```
$PCAN,C,GET,E:P0107P0113,*73<enter>
```

Jsou vráceny 2 chybové kódy: P0107 a P0113.

Význam kódů lze najít například zde: <http://www.obd-codes.com/>.

P0107 - Manifold Absolute Pressure/Barometric Pressure Circuit Low Input

P0113 - Intake Air Temperature Circuit High Input

Dotaz 2:

```
$PCAN,C,GET,#E<s3>##,*2D
```

Odpověď:

```
$PCAN,C,GET,E:U0073P0420,*71<enter>
```

```
$PCAN,C,GET,E:U0101U0155C0444,*30<enter>
```

```
$PCAN,C,GET,E:C0534C0446C2647C3700,*71<enter>
```

```
$PCAN,C,GET,E:P0011P0012,*75<enter>
```

V tomto případě přišlo několik odpovědí na dotaz na chybové kódy. Důvodem je buď skutečnost že některá řídicí jednotka obsahuje mnoho chybových kódů nebo odpovídá několik ŘJ.

Chybové kódy: U0073 P0420 U0101 U0155 C0444 C0534 C0446 C2647 C3700
P0011 P0012.

Power control

Volitelně může zákazník objednat chip obsahující funkci POWER CONTROL. V tomto případě je pin RE0 určen k připojení signálu zapalování nebo jiného řídicího signálu. Pin RE2 pak ovládá zdroj. Příslušné schéma zdroje a připojení signálu lze vyžádat.

Nadřazený systém může taktéž zapínat FMS OEM CHIP ovládáním pinu MCLR.

Specifikace objednávky

V objednávce je nutno specifikovat několik parametrů:

- 1) Bude-li chip v režimu SPI, lze volit pozitivní nebo negativní CS. Standardně negativní CS, tedy chip komunikuje je-li CS v low.
- 2) Je-li chip použit s SPI na nízké rychlosti CLK (například s GSM/GPS moduly Telit přes GPIO) je nutno toto taktéž specifikovat.
- 3) Zapnutá nebo vypnutá funkce Power control.
- 4) CAN po startu aktivní (nastavení ukládáno při změně do EEPROM) nebo neaktivní kdy je nutno vždy po startu zařízení nastavit (preferováno).
- 5) Na přání je možné chování chipu upravit, například přednastavit konkrétní nastavení během jeho programování.

Upozornění

Chip připojuje zákazník k vozidlu na **vlastní riziko**. Nesprávné nastavení chipu může způsobit nesprávnou funkci řídicích jednotek vozidla. Za škody na vozidle firma CANLAB s.r.o. neodpovídá.