

Při používání FMS OEM CHIPu V7 mohou nastat situace, kdy je třeba pracovat přímo s CAN sběrnici. Jsou to situace, kdy potřebujeme číst nestandardní data nebo data která FMS OEM CHIP neumí sám dekódovat a podobně. Následující příkazy jsou platné pro firmware od verze 2.50 včetně.

První možností je takzvaný SCAN mód. V tomto režimu FMS OEM CHIP odesílá surová CAN data po sériové lince do nadřazeného systému. Jelikož přenosová kapacita sběrnice je menší, než přenosová kapacita CAN sběrnice, jsou odesílána jen některá data. Vždy se zachytí část komunikace a ta se odešle do nadřazeného systému.

Pro aktivaci SCAN módu je určen tento příkaz:

```
$PCAN, C, SCN, 1, *30
```

V tomto případě se data i identifikátory odesílají dekadicky. Pokud chceme, aby byly v hexadecimálním formátu, je třeba použít tuto variantu příkazu:

```
$PCAN, C, SCN, 2, *33
```

Příklad výstupu dat při aktivaci SCAN módu pro 29 bitové identifikátory:

```
$PCAN, C, SCN, M15, E: FF022F, L8, FF, FF, FF, FF, FF, FF, FF, FF, *35  
$PCAN, C, SCN, M14, E: FF032F, L8, 10, E0, 03, 05, 80, DC, C0, 00, *3B  
$PCAN, C, SCN, M13, E: 18F0010B, L8, C0, FF, F0, FF, FF, 0D, FF, FF, *38  
$PCAN, C, SCN, M12, E: CF00300, L8, C1, 00, 1F, FF, FF, FF, FF, FF, *77  
$PCAN, C, SCN, M11, E: CFE6CEE, L8, FF, FF, FF, FC, FF, FF, 00, 00, *77  
$PCAN, C, SCN, M10, E: 1808013D, L8, 64, 2D, B8, 30, FA, FF, FF, C0, *48  
$PCAN, C, SCN, M9, E: C000F0B, L8, FC, FF, FA, 00, FF, FF, FF, FF, *3B  
$PCAN, C, SCN, M8, E: C00000B, L8, FC, FF, FA, FA, FF, FF, FF, FF, *4B  
$PCAN, C, SCN, M7, E: C07033D, L8, 72, 04, 00, 00, 00, 00, FF, FF, *41  
$PCAN, C, SCN, M6, E: C000003, L8, EC, FF, FF, FA, FF, FF, FF, FF, *30  
$PCAN, C, SCN, M5, E: 18FF2100, L8, 20, 11, 50, 00, 00, 00, 00, 56, *7C  
$PCAN, C, SCN, M4, E: CF00400, L8, F0, 7D, 8C, C8, 12, 00, F3, 8B, *4B  
$PCAN, C, SCN, M3, E: C000F03, L8, EF, FF, FF, 7D, FF, FF, FF, FF, *32  
$PCAN, C, SCN, M2, E: 18F0000F, L8, 50, 7D, 7D, FF, 00, 7D, FF, 7D, *79  
$PCAN, C, SCN, M1, E: 18F00503, L8, 7D, 00, 00, 7D, 20, 4E, 20, 4E, *0F  
$PCAN, C, SCN, M0, E: CF00203, L8, C0, 00, 00, FF, FF, AF, 11, FF, *37  
$PCAN, C, SCN, M8, E: CF00400, L8, F0, 7D, 8C, C8, 12, 00, F3, 8B, *47  
$PCAN, C, SCN, M7, E: CFF3D00, L8, 00, 00, F5, 60, 08, 59, 59, 40, *3D  
$PCAN, C, SCN, M6, E: C00100B, L8, FC, FF, FA, 00, FF, FF, FF, FF, *43  
$PCAN, C, SCN, M5, E: CF00203, L8, C0, 00, 00, FF, FF, BC, 11, FF, *34  
$PCAN, C, SCN, M4, E: 18FEDF00, L8, 83, 68, 01, 09, 7D, FF, FF, FF, *05  
$PCAN, C, SCN, M3, E: 18FF0021, L8, 00, FF, FF, FF, FF, FF, FF, FC, *7B
```

Příklad výstupu dat při aktivaci SCAN módu pro 11 bitové identifikátory:

```
<$PCAN, C, SCN, M6, S: 4C0, L6, 08, 00, 00, 09, D1, 00, *5A  
<$PCAN, C, SCN, M5, S: 312, L8, 08, 0A, 0D, 3A, 0A, 48, 07, 08, *59  
<$PCAN, C, SCN, M4, S: 212, L8, 03, 3E, A8, 12, 28, 12, A8, 12, *28  
<$PCAN, C, SCN, M3, S: 210, L8, 00, 00, 24, 40, 00, 00, 7E, 00, *21  
<$PCAN, C, SCN, M2, S: 268, L8, 00, 00, 00, 09, 00, 24, FF, FF, *50
```

```
<$PCAN,C,SCN,M1,S:308,L8,80,06,51,00,00,67,FF,FF,*50  
<$PCAN,C,SCN,M0,S:236,L8,0E,2F,0F,A6,34,FF,FF,27,*54  
<$PCAN,C,SCN,M8,S:328,L8,F7,FF,7F,00,DC,31,FB,40,*55  
<$PCAN,C,SCN,M7,S:258,L8,00,00,FF,FF,41,FF,00,00,*5C  
<$PCAN,C,SCN,M6,S:330,L6,84,80,89,30,00,00,*5F  
<$PCAN,C,SCN,M5,S:119,L8,00,00,00,00,00,03,09,2B,*27  
<$PCAN,C,SCN,M4,S:238,L8,01,1A,09,D1,FF,00,00,00,*51  
<$PCAN,C,SCN,M3,S:248,L8,00,00,00,00,30,00,00,00,*5F  
<$PCAN,C,SCN,M2,S:240,L8,00,12,00,00,00,C2,00,00,*27  
<$PCAN,C,SCN,M1,S:410,L8,11,09,24,00,00,66,00,00,*5A
```

V případě, kdy je třeba odesílat data na CAN sběrnici, je možné použít tyto příkazy:

Tento příkaz odešle na CAN sběrnici zprávu s 11 bitovým identifikátorem 123 dekadicky, počet datový bajtů je 5. První datový bajt je nastaven na hodnotu 11 dekadicky, druhý na hodnotu 22 dekadicky, ostatní mají hodnotu 0.

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

Je-li místo znaků CAN použita varianta CAX, je identifikátor i data možné zadat hexadecimálně. Délka identifikátoru může být proměnná dle potřeby. Datové bajty je však třeba uvádět vždy jako dvojici znaků. Tedy například pro datový bajt DB0 1h je třeba, aby příkaz obsahoval text B0:01

```
$PCAN,C,CAX,S,I123,S,L5,B0:11,B1:22,*28
```

Tyto příkazy jsou analogické k předchozím avšak s tím, že se jedná o variantu s 29 bitovým identifikátorem – písmeno E.

```
$PCAN,C,CAN,S,I123456,E,L5,B0:11,B1:22,*1F
```

```
$PCAN,C,CAX,S,I123456,E,L5,B0:11,B1:22,*09
```

Předchozí příklady odesílaly na CAN zprávu jednorázově. Existují ovšem situace, kdy se hodí, aby zpráva byla odesílána opakovaně periodicky. To je vhodné například v situaci, kdy je třeba data periodicky vyžadovat. Pro vyžadování je k dispozici až 8 bufferů. Příkaz je stejný jako předchozí, je zde však navíc uvedena perioda odesílání. V prvním příkladu, který následuje je uveden text P1. Jedná se o nastavení periody na násobek 1 základní periody 100ms. Zpráva je tak odesílána s periodou 100ms. Pro odesílání je použit buffer 0 – text BP0. Indexy bufferů jsou 0 až 7.

```
$PCAN,C,CAX,S,I123456,E,L5,B0:11,B1:22,P1,BP0,*4A
```

Tento druhý příklad nastaví zprávu do bufferu 0 a zpráva je odesílána s násobkem periody 2, tedy s periodou 200 ms.

```
$PCAN,C,CAX,S,I333,A,L5,B0:11,B1:22,P2,BP1,*78
```

Další situací, která může nastat, je potřeba číst konkrétní CAN zprávu. Pro to je určeno opět 8 bufferů.

Tento příkaz nastaví druhý buffer na příjem zprávy s 29 bitovým identifikátorem 1234 dekadicky.

```
$PCAN,C,CAN,T1,I1234,E,*7E
```

Chceme li přečíst, zda zpráva byla přijata a hodnoty datových bajtů, odešleme příkaz:

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

Pokud nebyla zpráva přijata, odpoví FMS OEM CHIP takto:

```
$PCAN, C, CAN, R1, E, *19
```

V případě, že přijata byla odpověď například takto:

```
$PCAN, C, CAN, R1, I1234, E, L8, B0:111, B1:125, B2:3, B3:4, B4:5, B5:6, B6:7, B7:8, *2C
```

Pokud se znovu zeptáme na zprávu-buffer R1 a ta nebyla od předchozího čtení z CANu zachycena, FMS OEM CHIP opět vrátí:

```
$PCAN, C, CAN, R1, E, *19
```

V některých případech je vhodné mít k dispozici filtraci nejen podle identifikátoru ale i podle datových bajtů. FMS OEM CHIP V7 podporuje filtraci podle hodnot prvních 4 datových bajtů.

Tento příklad ukazuje jak nastavit příjem zprávy s 11 bitovým identifikátorem o hodnotě 222 dekadicky. Zpráva je navíc přijata pouze v případě kdy druhý datový bajt (DB1) má hodnotu 10 dekadicky a třetí datový bajt pak hodnotu 2 dekadicky. Hodnota DB0 a DB3 se nebere při filtraci v úvahu. DB4 až DB7 nelze k filtraci použít.

```
$PCAN, C, CAN, T0, I222, S, B1:10, B2:2, *6F
```

Příklad odpovědi:

```
$PCAN, C, CAN, R0, I222, S, L8, B0:0, B1:10, B2:2, B3:40, B4:50, B5:60, B6:70, B7:80, *0A
```

Samozřejmě je možné použít i varianty pro nastavování a čtení v hexadecimálním formátu:

```
$PCAN, C, CAN, R0, IDE, S, L8, B0:00, B1:0A, B2:02, B3:28, B4:32, B5:3C, B6:46, B7:50, *3D
```

```
$PCAN, C, CAX, T0, I7DF, S, *4E
```

Třetí situací která nastává je případ, kdy některá data nejsou například u nákladních vozidel dle standardu SAE J939 ale například nádrž používá jiný PGN.

V tomto případě je možné nastavit vlastní dekodování dat:

```
$PCAN, R, L, 1, ID, 200, EXT, BIG, FI, 8, LE, 8, MU, 1.0, OF, 0, *01
```

L-nádrž

1-primární

ID – identifikátor CAN zprávy 200 dekadicky

EXT – 29 bitový identifikátor, je li standardní, neuvádí se

BIG – formát dat big endian

FI- první bit (LSB)

LE – počet datových bitů

MU – multiplikátor

OF-offset, odčítá se

Více lze najít v dokumentaci k FMS OEM CHIPu V7 v kapitole „Uživatelské dekódování CAN dat“.

Jak je vidět, funkce FMS OEM CHIPu nejsou limitovány pouze přeprogramovaným dekódováním dat, ale poskytují uživateli i poměrně velký prostor pro vlastní funkce. Verze firmware pro smluvní zákazníky navíc obsahují i další funkce pro přenos dat z CAN sběrnice a na CAN sběrnici. Jejich použití je ovšem možné pouze při odsouhlasení a podepsání licenčních podmínek použití a stanovení minimálního množství odběru. Taktéž cena za dodatečné funkce se řídí zvláštním ujednáním.