

Popis USB komunikace mezi PC a CAN bus adaptérem USB2CAN.

*Ing. David Španěl
CANLAB s.r.o.
spanel@canlab.cz*

I. Ovladače FTDI

Základem adaptéru USB2CAN je obvod firmy FTDI (www.ftdichip.com) FT245BM. Firma dodává 2 varianty ovladačů pro tento (a nejen tento) obvod. Jedna se o ovladače VCP a D2XX. API X2CAN je vystavěno na ovladačích D2XX. Důležité je upozornit, že současné ovladače nedovolují koexistenci obou verzí ovladačů na jednom systému zároveň. Ovladače jsou k dispozici na systémy W9x, WME, W2k a WXP. Dále jsou k dispozici pro operační systémy Linux, QNX, MAC OS a Windows CE.

API X2CAN je v současné době k dispozici pro operační systém Windows. Pokud chcete vytvořit vlastní variantu API pro některý z uvedených operačních systémů, je nutno znát strukturu komunikace po USB mezi PC a adaptérem USB2CAN. Stejně tak je tento dokument nutný pro případ, kdy experimentujete s vlastním firmware pro USB2CAN.

II. Struktura USB zprávy

Pro zasílání zpráv (příkazů, dat) mezi PC a USB2CAN adaptérem je nutno definovat vhodnou strukturu dat.

Byte:	0	1	2	3-18
Popis:	Start byte	Command	Length	Data
Hodnota:	0x0F	X	0x00 – 0x10	X

Prvním bajtem zprávy je takzvaný START_BYTE. Jeho hodnota je definována jako 0x0F. Tento bajt je určen pro synchronizaci přenosu mezi PC a adaptérem při jejím eventuálním přerušení. Následující bajt je označen jako COMMAND. Tímto bajtem sdělujeme PC nebo adaptéru o jaká data nebo příkaz se jedná. Od významu tohoto bajtu je pak odvozen význam dat v poli DATA. Pro jednotlivé příkazy COMMAND není z důvodu optimalizace přenosu mezi obvody PIC a FTDI stanovena pevná délka. Proto je uveden bajt LENGTH, který udává počet datových bajtů. Optimalizace přenosu se vyplatí při přenosu velkého množství CAN zpráv, které se mohou lišit svojí délkou. Počet datových bajtů se může pohybovat od 3, pro zprávu se standardním identifikátorem a nulovým počtem datových bajtů v CAN zprávě až po 13, pro zprávu s rozšířeným identifikátorem a 8 datovými bajty v CAN zprávě. Následující tabulka uvádí seznam podporovaných příkazů a datových zpráv.

Název	Hodnota	Mód	Popis
USB_LOOPBACK	0	B,C,N,L*	Loopback
BOOT_MODE	1	C,N,L	Přepnutí do Boot módu
CONFIG_MODE	2	B,N,L	Přepnutí do Config módu
NORMAL_MODE	3	C,L	Přepnutí do Normal módu
LOOPBACK_MODE	4	C,N	Přepnutí do Loopback módu
GET_MODE	6	B,C,N,L	Žádost o vrácení aktuálního módu
BUS_OFF	8	N	Hlášení adaptéru o stavu Bus-off
ERROR	9	N	Další chybová hlášení
READ_REG	16	C,N,L	Čtení registru SJA 1000
READ_REG_BLOCK**	17	C,N,L	Čtení bloku registrů SJA 1000
WRITE_REG	18	C,N,L	Zápis do registru SJA 1000
WRITE_REG_BLOCK**	19	C,N,L	Zápis do bloku registrů SJA 1000
WRITE_READ_REG	20	C,N,L	Zápis do registru SJA 1000 a zpětné čtení
BITMOD_REG	21	C,N,L	Bitová modifikace registru SJA 1000 pomocí masky
BITMOD_READ_REG	22	C,N,L	Bitová modifikace registru SJA 1000 pomocí masky a zpětné čtení
COMMAND	32	C,N,L	Rozšiřující příkazy
FW_VERSION	33	C,N,L	Žádost o vrácení verze firmware
TIMESTAMP****	62	N	Časová značka 1 sekunda
READ_MESSAGE_TS****	63	N	Příchozí CAN zpráva (USB2CAN->PC), poslední 2 bajty obsahují timestemp s rozlišením 250 mikrosekund s hodnotou v intervalu 1 sekunda (rozsah 0-3999).
WRITE_MESSAGE	64	N,L	Odeslání CAN zprávy
READ_MESSAGE	65	N,L	Příchozí CAN zpráva (USB2CAN->PC)
WRITE_SYNC_MESSAGE***	66	N	Zápis zprávy do tabulky HW synchronizačních zpráv.
READ_SYNC_MESSAGE***	67	N	Čtení zprávy z tabulky HW synchronizačních zpráv.
SEND_SYNC_MESSAGE***	68	N	Příkaz k odeslání zprávy z tabulky synchronizačních zpráv.
DISABLE_SYNC_MESSAGE***	69	N	Zakázání odesílání zprávy z tabulky.
DISABLE_ALL_SYNC***	70	N	Zakázání odesílání všech zpráv z tabulky.
ENABLE_SYNC_MESSAGE***	71	N	Povolení odesílání zprávy z tabulky.
SET_PERIOD_SYNC_MESSAGE***	72	N	Nastavení periody generování zprávy.
SUPPORTED_SYNC_MESSAGE	74	N	Dotaz a odpověď zda jsou HW synchronizační zprávy podporovány.
READ_TEC	96	N	Čtení Transmit Error Counteru
READ_REC	97	N	Čtení Receive Error Counteru
READ_RST	98	N	Počet restartů SJA1000 z důvodu přechodu do Bus-off stavu
WRITE_INSTRUCTION	127	B	Zápis instrukce do programové paměti

* B – BOOT MODE
C – CONFIG MODE
N – NORMAL MODE
L – LOOPBACK MODE

** Není implementováno

*** Jen pokud firmware podporuje HW synchronizační zprávy.

**** Jen pokud firmware podporuje funkci timestamp.

USB2CAN má 4 pracovní režimy: BOOT, CONFIG, NORMAL a LOOPBACK MODE. Sloupec Mode uvádí režim, ve kterém je zpráva podporována.

BOOT

V tomto režimu se firmware adaptéru nachází po připojení do USB PC. V tomto režimu je možno provádět změnu firmware adaptéru. Do tohoto režimu je možné se také dostat z režimu CONFIG zasláním příkazu BOOT_MODE.

CONFIG

V tomto režimu je obvod SJA přepnut do stejnojmenného režimu Config. Pouze v tomto módu je možné provádět nastavení timing registrů, filtrů zpráv apod.

NORMAL

V režimu normal je přepnut SJA 1000 do stejnojmenného režimu obvodu SJA 1000, je aktivován mód PeliCAN. V tomto módu adaptér přijímá zprávy z CAN sběrnice a akceptuje požadavky z PC na odeslání CAN zprávy na sběrnici.

LOOPBACK

Tento režim přijímá požadavky na odeslání CAN zprávy na sběrnici a tyto požadavky vrací zpět jakoby byla zpráva přijata z CANu.

Command:	Hodnota:	Mód:	Length:
USB_LOOPBACK	0	B,C,N,L	0

Příkaz je určen pro elementární kontrolu funkčnosti USB2CAN adaptéru a jeho USB rozhraní. Pole LENGTH má hodnotu 0 (nulový počet datových bajtů). USB2CAN na tuto zprávu odpovídá přeposláním této zprávy zpět. Na tuto zprávu reaguje firmware adaptéru ve všech režimech.

Command:	Hodnota:	Mód:	Length:
BOOT_MODE	1	C,N,	0

Příkaz je určen k přepnutí adaptéru do BOOT módu. Počet datových bajtů je nulový. Adaptér akceptuje tuto zprávu ve všech módech. Na tuto zprávu není generována adaptérem USB2CAN žádná odpověď.

Command:	Hodnota:	Mód:	Length:
CONFIG_MODE	2	B,C,N,L	0

Příkaz je určen k přepnutí adaptéru do CONFIG módu. Počet datových bajtů je nulový. Adaptér akceptuje tuto zprávu ve všech módech. Na tuto zprávu není generována adaptérem USB2CAN žádná odpověď.

Command:	Hodnota:	Mód:	Length:
NORMAL_MODE	3	C,L	0

Příkaz je určen k přepnutí adaptéru do NORMAL módu. Počet datových bajtů je nulový. Adaptér akceptuje tuto zprávu ve všech módech. Na tuto zprávu není generována adaptérem USB2CAN žádná odpověď. V tomto módu akceptuje adaptér požadavky na odeslání zprávy na CAN sběrnici. Stejně tak je to jediný mód, kdy je adaptér schopen přijímat zprávy s CAN sběrnice a zasílat je prostřednictvím USB do PC.

Command:	Hodnota:	Mód:	Length:
LOOPBACK_MODE	4	C,N	0

Příkaz je určen k přepnutí adaptéru do LOOPBACK módu. Počet datových bajtů je nulový. Adaptér akceptuje tuto zprávu ve všech módech. Na tuto zprávu není generována adaptérem USB2CAN žádná odpověď. V tomto režimu přijímá adaptér zprávy z USB s požadavky na odeslání CAN zprávy. Tyto zprávy však nejsou odeslány na CAN ale jsou odeslány zpět do PC jako přijatá zpráva z CANu. Jedná se o softwarovou obdobu módu LOOPBACK u PP2CAN adaptéru a obvodu MCP251x.

Command:	Hodnota:	Mód:	Length:
GET_MODE	6	B,C,N,L	0/1

Prostřednictvím této zprávy vyžaduje PC po adaptéru USB2CAN vrácení jeho aktuálního módu. Požadavek z PC má nulový počet datových bajtů. V odpovědi je 1 datový bajt. Jeho význam je následující:

Mód	Hodnota
BOOT	0
CONFIG	1
NORMAL	2
LOOPBACK	3

Command:	Hodnota:	Mód:	Length:
BUS_OFF	8	N	5

Detekuje – li adaptér na sběrnici CAN stav Bus Off, zašle do PC tuto zprávu. Zpráva obsahuje 6 datových bajtů, jednotlivé bajty obsahují data podle následující tabulky:

Datový bajt	Popis
0	Hodnota registru STATUS obvodu SJA 1000.
1	Hodnota registru Arbitration Lost Capture
2	Hodnota registru Error Code Capture
3	Hodnota registru Receive Error Counter
4	Hodnota registru Transmit Error Counter
5	Hodnota registru STATUS obvodu SJA 1000.

Následně je proveden reset obvodu SJA1000.

Command:	Hodnota:	Mód:	Length:
ERROR	9	N	X

Tato zpráva má přesný význam definován hodnotou prvního datového bajtu. V současné verzi jsou definovány tyto chybová hlášení:

Název	Hodnota	Popis
ERROR_CAN_TRANSMIT	0	Zpráva nemůže být odeslána, interní buffer procesoru PIC je plný. Zpráva je uložena v následujících 13 datových bajtech. Standardní X2CAN API zařadí tuto zprávu k opětovnému odeslání.
ERROR_STATUS	1	Zaslána do PC pokud dojde k nastavení bitu Error Status ve Status registru obvodu SJA 1000. Význam datových bajtů 1-5 (první datový bajt indexován jako 0) je shodný se zprávou BUS_OFF.
ERROR_CRITICAL_TRANSMIT	2	Zpráva je generována pokud dojde k překročení hranice Critical Transmit Limit. Tato hranice udává počet zpráv, které čekají na odeslání v PIC. Do PC je tak signalizováno s předstihem aby pozdržel zasílání dalších požadavků na odeslání dalších CAN zpráv. Při překročení této hranice je nastaven druhý datový bajt na 1. Jakmile dojde k poklesu počtu zpráv, které čekají na odeslání pod úroveň Ready Transmit Limit je tato zpráva generována znovu, druhý datový bajt je však nastaven na 0.
ERROR_RX_OVERRUN	3	Zpráva je zaslána do PC pokud dojde ke ztrátě zprávy při příjmu z CAN sběrnice vlivem nedostatku místa v přijímacím FIFO bufferu.

Command:	Hodnota:	Mód:	Length:
READ_REG	16	C,N,L	1/2

Zpráva je určena pro čtení registru SJA1000. PC zašle v této zprávě adresu požadovaného registru. Procesor PIC provede čtení a odešle tuto zprávu zpět. Adresa je uvedena v prvním datovém bajtu, hodnota registru v odpovědi pak v druhém datovém bajtu. Informaci o tom, které registry je možno číst, naleznete v datasheetu obvodu SJA1000. Obvod SJA 1000 pracuje v módu PeliCAN.

Command:	Hodnota:	Mód:	Length:
READ_REG_BLOCK	17	C,N,L	

Funkce není dosud implementována.

Command:	Hodnota:	Mód:	Length:
WRITE_REG	18	C,N,L	2

Tato zpráva je zasílána z PC do adaptéru USB2CAN. V prvním datovém bajtu obsahuje adresu, ve druhém pak hodnotu která má být zapsána na tuto adresu v obvodu SJA 1000. Adaptér zasílá tuto zprávu zpět do PC, aby potvrdil splnění požadavku. V odpovědi je uvedena pouze adresa registru. Delka odpovědi je tedy 1.

Command:	Hodnota:	Mód:	Length:
WRITE_REG_BLOCK	19	C,N,L	

Funkce není dosud implementována.

Command:	Hodnota:	Mód:	Length:
WRITE_READ_REG_	20	C,N,L	2/2

Tato zpráva je zasílána z PC do adaptéru USB2CAN. V prvním datovém bajtu obsahuje adresu, ve druhém pak hodnotu která má být zapsána na tuto adresu v obvodu SJA 1000. Adaptér zasílá tuto zprávu zpět do PC, aby potvrdil splnění požadavku. V odpovědi je na rozdíl od příkazu WRITE_REG obsažena i hodnota registru, která je získána zpětným čtením.

Command:	Hodnota:	Mód:	Length:
BITMOD_REG_	21	C,N,L	3/3

Příkaz, který nese tato zpráva je určen k modifikaci několika bitů zadaného registru obvodu SJA 1000. Ostatní bity zůstanou nezměněny. V tomto příkazu je tedy obsažena adresa registru, bitová maska pro výběr modifikovaných bitů a hodnota. Můžeme tak modifikovat jen některé bity registru, bez znalosti stavu ostatních bitů. Adresa je uvedena v prvním datovém bajtu, bitová maska ve druhém, hodnota ve třetím. Adaptér zasílá jako potvrzení zprávu zpět.

Command:	Hodnota:	Mód:	Length:
BITMOD_READ_REG_	22	C,N,L	3/4

Příkaz, který nese tato zpráva je určen k modifikaci několika bitů zadaného registru obvodu SJA 1000. Ostatní bity zůstanou nezměněny. V tomto příkazu je tedy obsažena adresa registru, bitová maska pro výběr modifikovaných bitů a hodnota. Můžeme tak modifikovat jen některé bity registru, bez znalosti stavu ostatních bitů. Adresa je uvedena v prvním datovém bajtu, bitová maska ve druhém, hodnota ve třetím. Na rozdíl od příkazu BITMOD_REG je v odpovědi doplněn čtvrtý datový bajt s hodnotou která je po ukončení modifikace v tomto registru obsažena.

Command:	Hodnota:	Mód:	Length:
COMMAND	32	C,N,L	1-x

Tato zpráva je určena pro zasílání rozšiřujících příkazů. Seznam a popis těchto příkazů je uveden v následující tabulce. Kód příkazu je uveden v prvním datovém bajtu.

Název	Hodnota	Popis
CMD_TRANSMIT_CRITICAL_LIMIT	0	V druhém datovém bajtu je obsažena hodnota s novým nastavením hodnoty Critical Transmit Limit. Maximální hodnota je pro současnou verzi firmware 18.
CMD_TRANSMIT_READY_LIMIT	1	V druhém datovém bajtu je obsažena hodnota s novým nastavením hodnoty Ready Transmit Limit. Maximální hodnota je pro současnou verzi firmware 17. Tato hodnota musí být vždy menší, než hodnota Critical Transmit Limit!
CMD_ENABLE_READ_TEC	2	Tento příkaz povoluje průběžné čtení registru TEC. Je-li čtení TEC zakázáno, je tento registr čten pouze při signalizaci BUS_OFF a ERROR_STATUS.
CMD_ENABLE_READ_REC	3	Tento příkaz povoluje průběžné čtení registru REC. Je-li čtení REC zakázáno, je tento registr čten pouze při signalizaci BUS_OFF a ERROR_STATUS.
CMD_BASIC_CAN	4	Příkaz provede přepnutí SJA1000 do módu BASIC_CAN. V tomto módu je však třeba provádět zpracování zpráv z PC pomocí příkazů READ_REG, WRITE_REG apod. Režim má význam pro ověřování některých postupů z PC bez zásahu do firmware. Přepnutí je možné pouze v CONFIG_MODU firmware adaptéru USB2CAN. Standardní obsluha při příchodu zprávy není funkční. Obsluhu je třeba emulovat taktéž z PC.
CMD_PELI_CAN	5	Příkaz provede přepnutí SJA1000 do módu PELI_CAN. V tomto módu je však třeba provádět zpracování zpráv z PC pomocí příkazů READ_REG, WRITE_REG apod. Režim má význam pro ověřování některých postupů z PC bez zásahu do firmware. Přepnutí je možné pouze v CONFIG_MODU firmware adaptéru USB2CAN. Standardní obsluha při příchodu zprávy není funkční. Obsluhu je třeba emulovat taktéž z PC.
CMD_RESET_MODE	6	Příkaz provede přepnutí SJA1000 do módu RESET_MODU. V tomto módu je však třeba provádět zpracování zpráv z PC pomocí příkazů READ_REG, WRITE_REG apod. Režim má význam pro ověřování některých postupů z PC bez zásahu do firmware. Přepnutí je možné pouze v CONFIG_MODU firmware.
CMD_OPERATING_MODE	7	Příkaz provede přepnutí SJA1000 do módu OPERATING_MODU. V tomto módu je však třeba provádět zpracování zpráv z PC pomocí příkazů READ_REG, WRITE_REG apod. Režim má význam pro ověřování některých postupů z PC bez zásahu do firmware. Přepnutí je možné pouze v CONFIG_MODU firmware.
CMD_BAUD_RATE	8	Zápis do Timin registrů SJA1000. Zápis je možný pouze v CONFIG_MODU adaptéru USB2CAN. Druhý datový bajt nese hodnotu registru Bus Timing 0, třetí bajt pak hodnotu registru Bus Timing 1.
CMD_GET	128	Vyžádání interních parametrů firmare USB2CAN adaptéru. Jedná se o tyto parametry: Critical Transmit Limit, Ready Transmit Limit, First TX

		buffer, Second TX buffer.
--	--	---------------------------

Command:	Hodnota:	Mód:	Length:
FW_VERSION	33	C,N,L	12

Tento příkaz slouží k vyžádání identifikátoru HW revize a verze firmware adaptéru USB2CAN. Datové pole odpovědi zaslané do PC má po převodu na textový řetězec například tento tvar: HW0002FW0004.

HW0002 odpovídá public revizi hardwaru, FW0004 pak udává public verzi firmware. Identifikační řetězec adaptéru má obecně tvar HWxxxxFWyzzz. Kód xxxx je určen pro identifikaci verze adaptéru. V současnosti se jedná o verzi 2, ta představuje revizi 1.2 adaptéru. Revize 1.0 a 1.1 byly vývojové vzorky.

Kód y identifikuje řadu firmware a kód zzz pak určuje verzi firmware v dané řadě.

Řada (kód Y)	Popis
0	Základní řada firmware. Optimalizována na množství přenesených dat.
1	Základní řada firmware doplněna o funkci HWSync. Tato funkce dovoluje specifikovat až 16 zpráv, které jsou generovány s rozlišením 1ms na CAN.
2	Speciální firmware určený k zatížení sběrnice vysokým množstvím přenášených dat. Tato funkce je používána při vývoji HW a FW zařízení pro sběrnici CAN i při vývoji FW pro USB2CAN.
3	Nová řada firmware. Tato verze nemá funkci HWSync, obsahuje však dvojnásobek vyrovnávací paměti pro přenos dat na CAN. Tento FW je optimalizován na množství dat přenesených z/na CAN bus.
4	Nová řada firmware. Obsahuje funkci HWSync.
5	Nová řada firmware. Tato verze nemá funkci HWSync, obsahuje však dvojnásobek vyrovnávací paměti pro přenos dat na CAN. Dále pak obsahuje funkci Timestamp pro měření času zachycení zprávy z CANu s přesností 250 mikrosekund.
6	Nová řada firmware. Obsahuje funkci HWSync. Dále pak obsahuje funkci Timestamp pro měření času zachycení zprávy z CANu s přesností 250 mikrosekund.
7	FMS firmware. FW provádí příjem dat z vozidlové CAN sběrnice a na dotaz poskytuje předzpracovaná data o vozidle.

Command:	Hodnota:	Mód:	Length:
WRITE_MESSAGE	64	N,L	3-x

Tato zpráva nese požadavek na odeslání zprávy na CAN. Struktura datové části zprávy odpovídá struktuře transmit bufferu obvodu SJA 1000. Tzn. nejprve hodnota registru, TX frame Information, dále pak 2 (standardní 11 bitový identifikátor) nebo 4 (rozšířený 29 bitový identifikátor) bajty registrů TX identifier a následně 0 až 8 datových bajtů CAN zprávy. Délka USB zprávy je tedy závislá na délce CAN zprávy.

Command:	Hodnota:	Mód:	Length:
READ_MESSAGE	65	N,L	3-x

Zpráva READ_MESSAGE obsahuje přijatou CAN zprávu a je zasílána po jejím přijetí adaptérem USB2CAN do PC. Struktura datových bajtů odpovídá struktuře receive bufferu obvodu SJA1000. Tzn. nejprve hodnota registru, RX frame Information, dále pak 2 (standardní 11 bitový identifikátor) nebo 4 (rozšířený 29 bitový identifikátor) bajty registrů RX identifier a následně 0 až 8 datových bajtů CAN zprávy. Délka USB zprávy je tedy závislá na délce CAN zprávy.

Command:	Hodnota:	Mód:	Length:
READ_TEC	96	N	

Touto zprávou zasílá PC požadavek na vrácení hodnoty registru TEC. Čtení hodnoty TEC musí být povoleno příkazem CMD_ENABLE_READ_TEC. Požadavek nemá žádná data, odpověď pak má pouze jeden datový bajt s hodnotou TEC.

Command:	Hodnota:	Mód:	Length:
READ_REC	97	N	0/1

Zpráva READ_REC zasílá PC požadavek na vrácení hodnoty registru REC. Čtení hodnoty REC musí být povoleno příkazem CMD_ENABLE_READ_REC. Požadavek nemá žádná data, odpověď pak má pouze jeden datový bajt s hodnotou REC.

Command:	Hodnota:	Mód:	Length:
READ_RST	98	N	0/2

Touto zprávou zasílá PC požadavek na vrácení hodnoty registru RST. Hodnota RST je typu unsigned int16. Udává počet resetu SJA 1000 při přechodu do Bus Off.

Command:	Hodnota:	Mód:	Length:
WRITE_INSTRUCTION	127	B	

Pro účely změny firmware je v bootlooaderu implementována zpráva nazvaná WRITE_INSTRUCTION. Zpráva má délku 4 bajtů. První dva bajty obsahují adresu instrukce, přičemž první bajt obsahuje dolních 8 bitů adresy. Druhá dvojice pak obsahuje 16-bitovou instrukci. Zápis je povolen pouze od adresy 0x500 do konce programové paměti.

III. Inicializace adaptéru

Inicializace adaptéru USB2CAN dopracovního režimu vyžaduje tyto kroky:

- 1) Nastavit konfigurační mód zasláním příkazu **CONFIG_MODE**.
- 2) Nastavit Reset mód obvodu SJA1000 zápisem hodnoty 0x01 do registru **Mode** (adresa 0). Použít příkaz WRITE_REG.
- 3) Nastavit registr **Clock divider** na hodnotu 0xC0. Použít příkaz WRITE_REG.
- 4) Nastavit filtry zpráv. Pokud nechceme filtrovat, je třeba nastavit registry **Acceptance code** na 0x00 a **Acceptance mask** na 0xFF. Použít příkaz WRITE_REG.
- 5) Nastavit registr **Output control** na hodnotu 0xDA. Použít příkaz WRITE_REG.
- 6) Nastavit registr **Interrupt enable** na hodnotu 0x03. Použít příkaz WRITE_REG.
- 7) Nastavit registry **Bus timing 0 a 1**. Obvod SJA1000 má připojen krystal na frekvenci 16MHz. Použít příkaz WRITE_REG.
- 8) Příkazem **COMMAND** nastavit parametry **CMD_TRANSMIT_CRITICAL_LIMIT** a **CMD_TRANSMIT_READY_LIMIT**.
- 9) Nastavit normal (operační) mód zasláním příkazu **NORMAL_MODE**.
- 10) Nastavit registr **Mode** (adresa 0). Podle použití filtrů zpráv, nastavit hodnotu 0x00 nebo 0x08. Použít příkaz WRITE_REG.

Od tohoto okamžiku adaptér akceptuje zprávy s požadavkem na odeslání CAN zprávy a zasílá zpět přijaté CAN zprávy.

Při ukončení práce s adaptérem doporučujeme zaslat tuto sekvenci příkazů:

- 1) Nastavit konfigurační mód zasláním příkazu **CONFIG_MODE**.
- 2) Nastavit Reset mód obvodu SJA1000 zápisem hodnoty 0x01 do registru **Mode** (adresa 0). Použít příkaz **WRITE_REG**.