

LIN protokol a jeho implementácia použitím Atmel AVR mikropočítačov

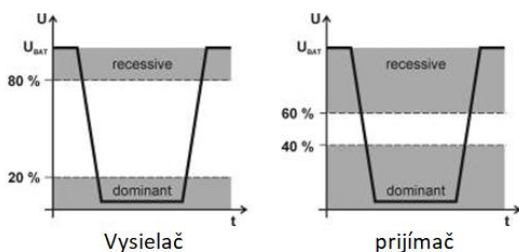
LIN protokol – bol navrhnutý špeciálne pre použitie v automobiloch

LIN – local interconnect network je protokol vytvorený v roku 1999 a prvý krát použitý v automobile v roku 2001. Oproti CAN zbernici je výhodou jednoduchosť a ekonomická stránka, pretože komunikácia zariadení cez LIN protokol môže byť 2x lacnejšia oproti CAN zbernici.

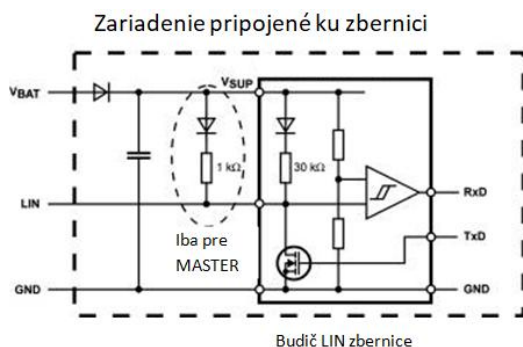
LIN je sériová asynchrónna zbernica používaná na jednovodičové spojenie pripojených zariadení, slúži na prepojenie zariadení, ktoré nepotrebujú komplikovanejšiu CAN zbernici. LIN zbernica dokáže preniesť menej informácií (20kbaud), ako CAN zbernica (1Mbaud). Používa sa predovšetkým na ovládanie a polohovanie spätných zrkadiel, sťahovanie okien, ovládanie zámkov dverí a strešného okna, polohovanie sedadiel, ovládanie klimatizácie, stieračov, alebo osvetlenia. Zariadenia sú prístupné pre všetky typy diagnostiky pri servisných prácach.

Princíp komunikácie je single master/multiple slave, čo znamená, že jedno riadiace zariadenie kontroluje komunikáciu s viacerými podriadenými. Maximálna prenosová rýchlosť je 20kbit/s. Podriadené zariadenia nepotrebujú presný kryštálový generátor hodín, ale stačí im bežný RC oscilátor, čo umožňuje zmenšiť cenu daných zariadení.

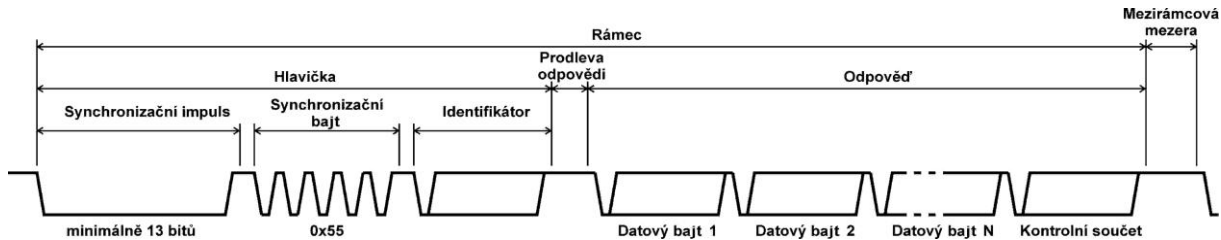
Fyzická vrstva je navrhnutá tak, že strmosť nábežných a dobežných hrán je zámerne obmedzená kvoli minimalizácii vyžarovaného rušenia. Zmenené sú tiež rozhodovacie úrovne s ohľadom na spoľahlivosť komunikácie. Princíp prenosu pomocou jedného vodiča je realizovaný pomocou realizácie logickej funkcie AND prostredníctvom spínačov a rezistorov zapojených na LIN zbernici v každom pripojenom zariadení. Hodnoty rozhodovacích napätí - recessive a dominant sú odvodené relatívne od napájacieho napätia z batérie auta podľa obrázku:



Maximálna možná dĺžka zbernice je 40m. Maximálny počet zariadení pripojených na zbernici by nemal prekročiť 16. Budič LIN zbernice je súčasťou pripojeného zariadenia a je konštruovaný tak, aby odolával rôznym poruchovým stavom, ako je napríklad usmerňovacia dióda na vývodoch budiča a kondenzátor pre elimináciu elektromagnetického rušenia:



Rámec správy pre LIN zbernicu je jednoduchý, slúži ku synchronizácii, adresácii a výmene dát medzi zariadeniami. Riadiaca jednotka začína komunikáciu, určuje prenosovú rýchlosť, a vysiela hlavičku rámca správy. Hlavička sa začína synchronizačným impulzom a následným synchronizačným poľom. Toto pole slúži k synchronizácii pripojených zariadení a synchronizácii prenosovej rýchlosti. Odpoveď je zložená z dátových Byte-ov a kontrolného súčtu:



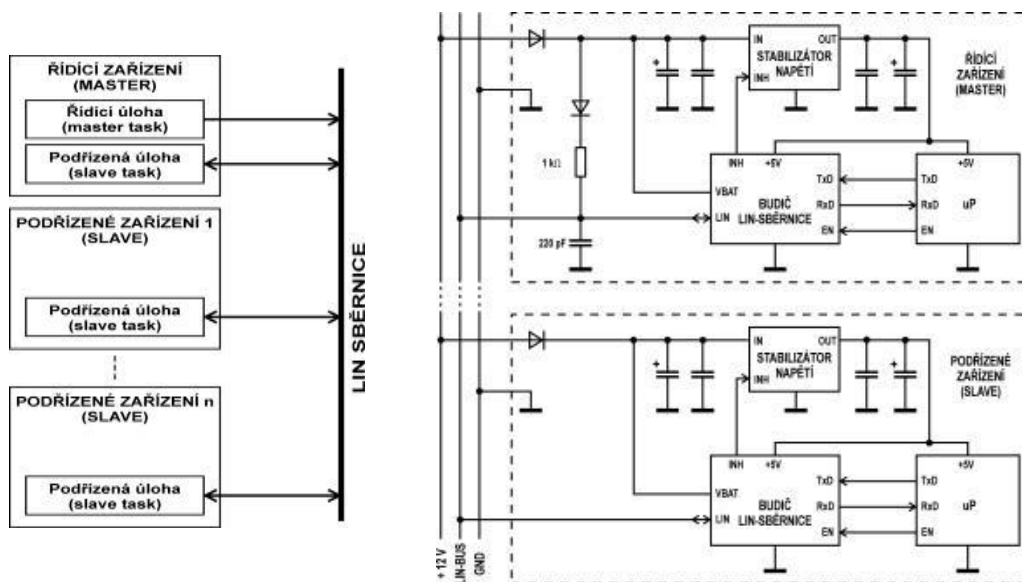
Funkcie riadiaceho zariadenia:

Riadenie komunikácie na zbernici, definuje prenosovú rýchlosť, vysiela synchronizačný impulz, synchronizačné pole, identifikátor, monitoruje a potvrdzuje dáta pomocou kontrolného súčtu, prepína podriadené jednotky do sleep módu a znova ich prebúdzá, reaguje na signál prebudenia od slave zariadení

Funkcie podriadeného zariadenia:

Čakanie na synchronizačné pole, synchronizácia podľa synchronizačného poľa, podľa identifikátora vykoná úlohu(nereaguje/prijíma dáta/vysiela dáta/kontroluje alebo posíela kontrolný súčet)

Príklad komunikačnej štruktúry:



Implementácia mikroprocesorov AVR do LIN protokolu

AVR - Vyvinutý nórskou pobočkou firmy Atmel koncom deväťdesiatych rokov ako jeden z mála nových mikropočítačov posledných dvoch desaťročí. Aj keď ide o 8-bitové jadro, bolo vyvíjané s ohľadom na ľahkú programovateľnosť v jazyku C a nesie znaky procesorov typu RISC. Vďaka pomerne vysokému výkonu, pomerne veľkému množstvu variánt s množstvom zabudovaných periférií a priaznivej cene pri nízkych objemoch nadobudol veľkú popularitu.

Zoznam mikropočítačov podporujúcich LIN protokol:

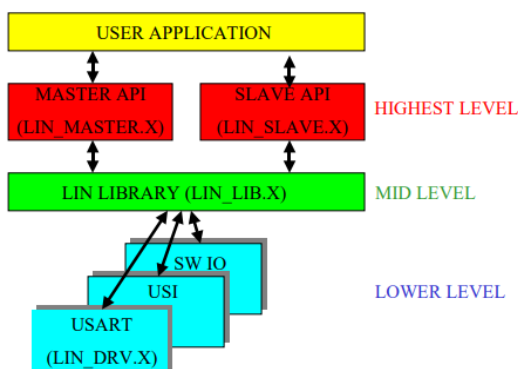
ATMEGA48/88/168

AT90CAN128

ATTINY45/85 – iba Slave

Pre aplikáciu LIN protokolu je vnútorný oscilátor v mikropočítačoch dôležitou vlastnosťou a jeho zabudovanie do mikropočítaču vedie ku zníženiu výrobných nákladov pre celý systém pripojený na LIN zbernici, presnosť zabudovaného oscilátora je $\pm 1\%$, čo je postačujúce pre fungujúcu komunikáciu na LIN zbernici, pretože LIN zbernica vyžaduje presnosť aspoň $1,5\%$. Toto však platí iba pre zariadenia Slave, pretože požadovaná presnosť pre Master zariadenie je $\pm 0,5\%$, čo znamená, že pre Master zariadenie musíme použiť externý oscilátor s vyhovujúcou presnosťou.

Pre implementáciu protokolu LIN sa rozlišujú tri software vrstvy:



Low level vrstva je prepojená s danou perifériou (USART, USI). Volaná je LIN knižnicou, teda strednou vrstvou. Hlavnou funkciou je ovládanie vybraného zariadenia na LIN zbernici, aktualizácia ovládača LIN, hlásenie chýb v rámcoch alebo bitoch do vyšších vrstiev, sprostredkovanie a prenos dát, generovanie/kontrola kontrolného súčtu, detekcia a synchronizovanie so synchronizačným Byte-om.

Mid level vrstva obsahuje inštrukcie pre funkcie a premenné vyskytujúce sa v LIN protokole. Úlohou mid level vrstvy je inicializácia budiča LIN (baud rate), prepínanie medzi wakeup/sleep módmí, interpretácia daných úloh pre zariadenia pripojené ku zbernici LIN.

High level vrstva obsahuje rozhranie pre programovanie aplikácií (API), ide o programové celky, ktoré určujú akým spôsobom sa majú funkcie knižníc volať zo zdrojového kódu programu. Hlavné úlohy sú zaobchádzanie s dátovými buffermi (s dátami), zabezpečovanie komunikácie priradením identifikátora k danému zariadeniu, počítadlo chýb, ovládanie činnosti počas sleep módu (žiadna aktivita na zbernici), uistenie sa, že Slave je pripravený prijať nové informácie.

Príkazy pre nastavenie zariadenia Master:

Zadanie LIN knižnice "MASTER_LIN.H" a jej konfigurácie "CONFIG.H"

Deklarácia dátových bufferov a ich inicializácia: `U8 Buf_SET_SLAVE [8];`

Deklarácia a inicializácia dátových rámcov

```
t_frame MESS_SET_SLAVE;
```

```
MESS_SET_SLAVE.frame_id = SET_SLAVE_FRAME_ID ;
```

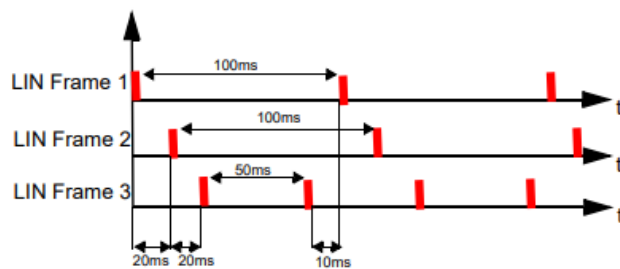
```
MESS_SET_SLAVE.frame_size = SET_SLAVE_FRAME_DLC ;
```

```
MESS_SET_SLAVE.frame_type = SET_SLAVE_FRAME_TYPE ;
```

```
MESS_SET_SLAVE.frame_delay = 1500 ; // delay in tick
```

```
MESS_SET_SLAVE.frame_data = Buf_SET_SLAVE;
```

Veľmi dôležité je načasovanie jednotlivých rámcov, napríklad ako na obrázku, aby sa komunikácia navzájom nebila:



Na to je príkaz ($delay_tick = 20ms * 19200baud = 384$)

Jednotlivé rámce a ich číslovanie možno napláňovať nasledovne:

```
my_schedule.frame_message[0] = MESS_SET_SLAVE;
```

```
my_schedule.number_of_frame = number_of_frame;
```

Inicializácia LIN sa nastaví príkazom `LIN_init()`, ktorá sa musí vykonať v hlavnej (main) funkcii po inicializácii vstupných/výstupných portov: `CONFIG_IO_PORTS()`

Ďalším krokom je povolenie prerušení `SREG |= 0x80;`

Tieto príkazy zabezpečujú inicializáciu a konfiguráciu Master zariadenia v LIN zbernici

Príkazy pre nastavenie zariadenia Slave:

Zadanie LIN knižnice "SLAVE_LIN.H" a jej konfigurácie "CONFIG.H"

Deklarácia a inicializácia dátových rámcov – nastaviť buffre potrebné pre aplikáciu LIN

```
t_frame MESS_SET_SLAVE;
```

```
MESS_SET_SLAVE.frame_id = SET_SLAVE_FRAME_ID ;
```

```
MESS_SET_SLAVE.frame_size = SET_SLAVE_FRAME_DLC ;
```

```
MESS_SET_SLAVE.frame_type = SET_SLAVE_FRAME_TYPE ;
```

```
MESS_SET_SLAVE.frame_data = Buf_SET_SLAVE;
```

Jednotlivé rámce a ich číslovanie sa nastavujú podobne ako pre Master:

```
my_schedule.frame_message[0] = MESS_SET_SLAVE;
```

```
my_schedule.number_of_frame = number_of_frame;
```

Inicializácia LIN sa nastaví príkazom *LIN_init()*, ktorá sa musí vykonať v hlavnej (main) funkcii po inicializácii vstupných/výstupných portov: *CONFIG_IO_PORTS()*

Ďalším krokom je povolenie prerušení *SREG |= 0x80;*

Tieto príkazy zabezpečujú inicializáciu a konfiguráciu Slave zariadenia v LIN zbernici

Software/hardware náročnosť daného riešenia:

Dané nastavenia pre Master/slave zariadenia využívajú približne 4kB miesta pre kód a 648B pre dáta bez optimalizácie a kód je možné optimalizovať na veľkosť 3.1kB. V najhoršom prípade je mikroprocesor zaťažený na 5% pri najvyššom baud rate 19200 a najmenej je zaťažený pri najmenšom baud rate 2400 a to na 0,6%.

Záver:

Opísané operácie poskytujú software-ové riešenie pre implementáciu LIN protokolu pomocou Atmel AVR mikropočítačov