

Канал ввода-вывода шины CAN

СОДЕРЖАНИЕ

1. Общие сведения.....	1
2. Конфигурации канала в RESIDENT.CNF.....	1
3. Программный интерфейс драйвера.....	2

1. Общие сведения

В многозадачное ядро «RESIDENT» введена поддержка канала ввода-вывода с интерфейсом CAN.

Поддерживаются контроллеры *:

- Intel CAN82C200;
- Philips SJA1000;

* - тестирование проводилось на плате встраиваемого промышленного компьютера Fastwel CPC105.

Драйвер канала обеспечивает:

- буферизованный ввод/вывод (программный, аппаратный);
- возможность назначения скорости работы шины;
- режимы передачи сообщений: синхронный, асинхронный;
- обработку ошибок.

Для корректной работы необходимо установить скорость канала равной скорости других устройств в шине. При работе с этим типом канала следует помнить что сообщения, передаваемые в шину, принимаются всеми устройствами, находящимися в шине.

2. Конфигурации канала в RESIDENT.CNF

Строка, описывающая драйвер канала ввода-вывода в файле RESIDENT.CNF имеет следующий вид:

DEVICE = CAN82C200,[IRQn],[BaseMem],[IBuffSize],[OBuffSize]

- | | |
|------------------|--|
| IRQn | - номер аппаратного прерывания, используемого CAN-контроллером, шестнадцатеричное число (по умолчанию Ah); |
| BaseMem | - базовый адрес разделяемого сегмента памяти, шестнадцатеричное число (по умолчанию 0DF00Hh); |
| IBuffSize | - размер входного буфера в байтах, шестнадцатеричное число (по умолчанию 100h / 256 байт); |
| OBuffSize | - размер выходного буфера в пакетах (посылка), шестнадцатеричное число (по умолчанию 10h / 16 пакетов); |

Пример:

```
DEVICE = CAN82C200,A,DF00,100,10  
DEVICE = CAN82C200,B,DF20,200,100
```

3. Программный интерфейс драйвера

Для работы с каналом ввода-вывода шины CAN используются стандартные вызовы многозадачного ядра «RESIDENT» `f_ttyread`, `f_ttywrite` и `f_ttycntl`, описанные в файле "tty.h".

Для конфигурирования канала во время работы программы используются функция:

int f_ttycntl (int n_can_dev, int func, int param);

n_can_dev - номер канала ввода-вывода образованного драйвером (0...);

func - команда драйверу канала;

param - аргумент команды (если требуется);

Прототип функции описан в "tty.h", определения значений аргументов `X_CAN_xxx` в "can.h".

Команда ***func*** на изменение параметров канала, может принимать одно из следующий значений:

`X_COM_PARAM` - изменение скорости, `param` должен содержать значение устанавливаемой скорости:

`X_CAN_10` - 10KBit/sec, максимальное расстояние 6000м

`X_CAN_125` - 125KBit/sec, максимальное расстояние 500м

`X_CAN_250` - 250KBit/sec, максимальное расстояние 200м

`X_CAN_500` - 500KBit/sec, максимальное расстояние 100м

`X_CAN_1000` - 1000KBit/sec, максимальное расстояние 40м

`X_RESET` - переинициализировать CAN контроллер (сбросить флаги ошибок, очистить буферы)

`X_CAN_TRANSMODE` - изменить режим посылки сообщений, `param` должен содержать одно из значений:

`X_CAN_SYNCH_TRANS` - для синхронной передачи

`X_CAN_ASYNC_TRANS` - для асинхронной передачи

Пример инициализации канала:

/ инициализируем канал 0, скорость 500KBit/sec, синхронный режим передачи */*

f_ttycntl(0, X_COM_18TIME, TIME_OUT);

f_ttycntl(0, X_COM_PARAM, X_CAN_500);

f_ttycntl(0, X_CAN_TRANSMODE, X_CAN_SYNCH_TRANS);

Прием и передача производится при помощи функций:

int f_ttyread (int n_can_dev, char *buff, int count);

int f_ttywrite (int n_can_dev, char *buff, int count);;

n_can_dev - номер канала ввода-вывода образованного драйвером (0...);

buff - указатель на буфер;

count - длина принимаемого или передаваемого сообщения в байтах.

Пример приема и передачи:

```
char buffer[10];
struct MSG_STRUCT msg;
int i, post_cnt;

post_cnt = 5;

msg.id0 = 0xbe;
for (i=0; i<post_cnt; i++) msg.data[i] = i;

f_ttywrite(can_dev1, (char*)&msg, post_cnt+2);
f_ttyread(can_dev2, buffer, post_cnt+2);
```