**Блок** Init\_Can(1|2) **инициализации CAN(1|2) контроллера**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да/нет) |

Блок реализует настройку контроллера CAN(1 |2) интерфейса. Программная реализация блока, расположена в заголовочном файле can.h файла main.c проекта Test\_Driver (в среде разработки Keil), который создается при генерации кода проекта и содержит тексты программ алгоритма на языке СИ. Блок настройки порта реализован на основании блока язык программирования и содержит в себе код на встроенном языке Simentech формирующем значения входных переменных для функции настройки порта Init\_Can(1|2) ().

Всего у микросхемы K1986ВЕ92QI, помещенной в корпус LQFP64, есть два контроллера CAN1 и CAN2, поэту библиотека содержит два блока для настройки контроллеров Init\_Can1 и Init\_Can2.

**Входные порты**

**Init\_port** соединяется с выходом блока настройки порта (необходимо определить выводы порта для приема и передачи пакетов с сообщениями RX, TX)

**Выходные порты**

**ready**  - Возвращает 1, это необходимо для определения порядка сортировки последущих блоков ;

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Прием собственных пакетов** | **ROP** | Прием собственных пакетов (Receive own packets):  1 – контроллер принимает собственные пакеты;  0 – контроллер принимает только чужие пакеты. | ENABLE |
| **Подтверждение собственных пакетов** | **SAP** | Подтверждение собственных пакетов (Send ACK on own packets):  1 – контроллер подтверждает прием собственных пакетов;  0 – контроллер подтверждает прием только чужих пакетов | ENABLE |
| **Режим самотестирования (Self Test Mode)** | **STM** | Режим самотестирования (Self Test Mode):  1 – контроллер работает в режиме самотестирования;  0 – контроллер работает в нормальном режиме | ENABLE |
| **Режим «Только прием» (Read Only Mode)** | **ROM** | Режим «Только прием» (Read Only Mode):  1 – контроллер работает только на прием;  0 – контроллер работает в нормальном режиме | DISABLE |
| **Значение фазы PSEG** | **PSEG** | Propagation Time Segment  Эта часть предназначена, чтобы компенсировать физические задержки времени  распространения сигнала в шине и внутренние задержки в узлах. Длительность этого интервала  может быть запрограммирована от 1 до 8 TQ | CAN\_PSEG\_Mul\_2TQ |
| **Значение фазы SEG1** | **SEG1** | Phas e Buf f er Segments  Эти интервалы предназначены для более точной установки точки семплирования, которая  располагается между ними. Длительности этих интервалов могут быть запрограммированы  между 1 и 8 TQ.  Значение размера фазы SEG1:  111 = Phase Segment 1 time = 8 x TQ  110 = Phase Segment 1 time = 7 x TQ  101 = Phase Segment 1 time = 6 x TQ  100 = Phase Segment 1 time = 5 x TQ  011 = Phase Segment 1 time = 4 x TQ  010 = Phase Segment 1 time = 3 x TQ  001 = Phase Segment 1 time = 2 x TQ  000 = Phase Segment 1 time = 1 x TQ  SEG1 – это время, используемое для увеличения битового  интервала при подстройке | CAN\_PSEG\_Mul\_3TQ |
| **Значение фазы SEG1** | **SEG2** | Phase Buf f er Segments  Эти интервалы предназначены для более точной установки точки семплирования, которая  располагается между ними. Длительности этих интервалов могут быть запрограммированы  между 1 и 8 TQ.  Значение размера фазы SEG2:  111 = Phase Segment 2 time = 8 x TQ  110 = Phase Segment 2 time = 7 x TQ  101 = Phase Segment 2 time = 6 x TQ  100 = Phase Segment 2 time = 5 x TQ  011 = Phase Segment 2 time = 4 x TQ  010 = Phase Segment 2 time = 3 x TQ  001 = Phase Segment 2 time = 2 x TQ  000 = Phase Segment 2 time = 1 x TQ  SEG2 – это время, используемое для сокращения битового  интервала при подстройке | CAN\_PSEG\_Mul\_2TQ |
| **Семплирование** | **SB** | Семплирование:  0 – однократное;  1 – трехкратное с мажоритарным контролем | CAN\_SB\_1\_SAMPLE |
| **Значение размера фазы SJW** | **SJW** | Значение размера фазы SJW:  11 = Synchronization jump width time = 4 x TQ  10 = Synchronization jump width time = 3 x TQ  01 = Synchronization jump width time = 2 x TQ  00 = Synchronization jump width time = 1 x TQ  SJW – это максимальное значение на которое происходит  подстройка приема и передачи при работе на шине CAN.  Приемник подстраивается на значение ошибки, но не более чем  SJW | CAN\_PSEG\_Mul\_4TQ |
| **Предделитель системной частоты** | **BRP** | Предделитель системной частоты:  CLK = PCLK/(BRP + 1)  TQ(us) = (BRP + 1)/CLK(MHz) | 1 |

**Пример Can\_Send.prt, Can\_Rcv.prt, Can\_Send\_Rcv.prt**

**Блок CAN\_Transmit (1|2) посылка пакета по CAN сети**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да/нет) |

Блок использует функцию CAN\_Transmit () из библиотеки MDR32F9Qx\_StdPeriph\_Driver и посылает по сети пакет данных.

**Входные порты**

**Initcan** порт подключения выхода блока инициализации CAN контроллера;

**Data0** значение посылки размером uint32\_t

**Data1** значение посылки размером uint32\_t

**Выходные порты**

**Ready** посылка произошла

**Status** значение статуса CAN контроллера

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Тип идентификатора** | **CAN\_ID** | Идентификатор – стандартный формат . Длина идентификатора - 11 бит и  соответствует Standart ID в расширенном формате. Эти биты передаются в порядке Bit28 …  Bit18. Самый младший бит - Bit18. 7 старших бит (Bit28 - Bit 22) не должны быть все  единичными битами.  Идентификатор – расширенный формат . В отличие от стандартного  идентификатора, расширенный идентификатор состоит из 29 бит. Его формат содержит две  секции:   Standart ID - 11 бит   Extended ID - 18 бит  Standar t I D состоит из 11 бит. Эта секция передается в порядке от Bit28 … Bit18. Это  эквивалентно формату стандартного идентификатора. Standart ID определяет базовый  приоритет расширенного пакета.  Ex tended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В  стандартном пакете идентификатор сопровождается RTR битом. | CAN\_ID\_STD (стандартный) |
| **Приоритет сообщения** | **PRIOR\_0** | Если у сообщения бит PRIOR\_0 выставлен в ноль, то оно отправляется в первую  очередь. | DISABLE |
| **Идентификатор** | **ID** | Standar t I D состоит из 11 бит. Эта секция передается в порядке от Bit28 … Bit18. Это  эквивалентно формату стандартного идентификатора. Standart ID определяет базовый  приоритет расширенного пакета.  Ex tended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В  стандартном пакете идентификатор сопровождается RTR битом. | нет |
| **Номер буфера** | **NUM\_BUF** | Всего 32 буфера для отправки и приема сообщений | нет |
| **Размер передаваемых**  **данных** | **DLC** | Размер передаваемых  данных кодируется в поле DLC и может составлять от 0 до 8 байт | нет |

**Пример Can\_Send.prt, Can\_Send\_Rcv.prt**

**Блок CAN(1|2)\_RCV приема пакетов данных**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да/нет) |

Блок использует прерывания процессора при получении пакета данных по сети CAN.

**Входные порты**

**Initcan** порт подключения выхода блока инициализации CAN контроллера

**Выходные порты**

**Data0** данные полученные по сети CAN размер uint32\_t

**Data1** данные полученные по сети CAN размер uint32\_t

**Dlc** размер данных в пакете, максимально размер Data0 + Data1,

Может быть равен 8 байт

**Newdata** флаг имеет значение 1 в том случае если пришел пакет с данными

0 если данных нет

**Can\_status** статус CAN контроллера

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Перезапись приемного буфера** | **OVER\_WRITE** | В буфере может быть включено разрешение перезаписи принятого сообщения. Если  принимаемое сообщение не может быть сохранено в свободный буфер, то оно может быть  сохранено в буфер с ранее полученным сообщением, если для него выставлен бит OVER\_EN.  При этом выставляется флаг OVER\_WR. Таким образом, если у буфера разрешена перезапись  принятых сообщений, после прочтения сообщения необходимо проверить флаг OVER\_WR.  Если он выставлен в 1, то необходимо сбросить OVER\_WR (не сбрасывая флаг RX\_FULL),  затем еще раз прочесть сообщение, после чего снова проверить флаг OVER\_WR и, если он не  выставлен повторно, то сбросить флаг RX\_FULL. И считанное значение считать корректным. | DISABLE |
| **Фильтр ID** | **FILTER\_ID** | Для уменьшения затрат процессорного ядра на обработку принимаемых сообщений,  контроллер CAN интерфейса может автоматически фильтровать принимаемые сообщения | 0 |
| **Маска** | **MASK\_ID** | Вторая часть фильтра | 0 |
| **Номер буфер** | **NUM\_BUF** | Номер буфера для приема пакета с данными | 0 |

**Пример Can\_Send\_Rcv.prt, Can\_Rcv.prt**

**Блок CAN(1|2)\_SendRTR посылка пакета запроса**

|  |  |  |
| --- | --- | --- |
| Картинка блока в палитре | Картинка блока на схеме | Векторизован ли блок, генерится ли блок в СИ код (да/нет) |

Блок посылает пакет удаленного запроса данных , чтобы запросить передачу пакета с данными

**Входные порты**

**Initcan** порт подключения выхода блока инициализации CAN контроллера

**Id** идентификатор в зависимости от типа пакета:

- Standart ID - 11 бит

- Extended ID - 18 бит

**Выходные порты**

Redy Блок отработал

Can\_status статус CAN контроллера

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Тип идентификатора** | **CAN\_ID** | Идентификатор – стандартный формат . Длина идентификатора - 11 бит и  соответствует Standart ID в расширенном формате. Эти биты передаются в порядке Bit28 …  Bit18. Самый младший бит - Bit18. 7 старших бит (Bit28 - Bit 22) не должны быть все  единичными битами.  Идентификатор – расширенный формат . В отличие от стандартного  идентификатора, расширенный идентификатор состоит из 29 бит. Его формат содержит две  секции:   Standart ID - 11 бит   Extended ID - 18 бит  Standar t I D состоит из 11 бит. Эта секция передается в порядке от Bit28 … Bit18. Это  эквивалентно формату стандартного идентификатора. Standart ID определяет базовый  приоритет расширенного пакета.  Ex tended ID состоит из 18 бит. Эта секция передается в порядке от Bit17 до Bit0. В  стандартном пакете идентификатор сопровождается RTR битом. | CAN\_ID\_STD (стандартный) |
| **Приоритет сообщения** | **PRIOR\_0** | Если у сообщения бит PRIOR\_0 выставлен в ноль, то оно отправляется в первую  очередь. | DISABLE |
| **Номер буфера** | **NUM\_BUF** | Всего 32 буфера для отправки и приема сообщений | нет |