Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Компьютерные системы и сети»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ НА ТЕМУ:

Исследование интерфейсов взаимодействия учебной платы Миландр

по дисциплине «Микропроцессорные системы»

Студент ИУ6-73 _____ **Ю.А. Шашкин**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ **В.Я. Хартов**
(Подпись, дата) (И.О.Фамилия)

Москва, 2019 г.

РЕФЕРАТ

Записка 49 стр., 7 таб., 25 рис., 12 источников, 2 прил.

МИКРОКОНТРОЛЛЕР, ФАЙЛ, ЗАПИСЬ, ДИСПЛЕЙ, КАРТА, ТАЙМЕР

Разработано что-то там

Материалы по курсовой работе представлены в виде графической части, приложения с отлаженным программным кодом для микроконтроллера и расчетно-пояснительной записки.

СПИСОК СОКРАЩЕНИЙ

МК – микроконтроллер

АЦП – аналого-цифровой преобразователь

ЦАП – цифро-аналоговый преобразователь

ОЗУ – оперативное запоминающее устройство

LCD – liquid crystal display (жидкокристаллический дисплей)

СОДЕРЖАНИЕ

Оно будет

ВВЕДЕНИЕ

С развитием технологий разработки микросхем, сложные вычислительные задачи могут быть решены с помощью микроконтроллеров, имеющих очень компактные размеры.

АО "ПКК Миландр" — ведущий российский разработчик и производитель изделий микроэлектроники (микроконтроллеры, микропроцессоры, микросхемы памяти, микросхемы приемопередатчиков, микросхемы преобразователей напряжения, радиочастотные схемы), универсальных электронных модулей и приборов промышленного и коммерческого назначения, разработки ПО для современных информационных систем и изделий микроэлектроники. Устройства данной компании по праву пользуются достаточно высокой популярностью на российском и зарубежном рынке.

Цель работы – исследование интерфейсов взаимодействия двух отладочных плат для микросхемы К1986ВЕ92QI (ТСКЯ.469575.002-01 вер. 4) и разработка тестирующих программ для модулей SSP, UART, CAN и USB. Разработка программ взаимодействия двух микроконтроллеров, демонстрирующих работу перечисленных интерфейсов, включая средства для отображения приема/передачи данных по интерфейсам.

1 Конструкторская часть

1.1 Краткое описание архитектуры и характеристик К1986ВЕ92QI

На основе информации с официального сайта АО "ПКК Миландр" со страницы данного МК может быть получена информация о его основных технических характеристиках. Данная информация является общей для всех микроконтроллеров серии 1986ВЕ9х.

1) Ядро:
   - ARM 32-битное RISC-ядро Cortex™-M3 ревизии 2.0, тактовая частота до 80 МГц,
   - производительность 1.25 DMIPS/МГц (Dhrystone 2.1) при нулевой задержке памяти;
   - блок аппаратной защиты памяти MPU;
   - умножение за один цикл, аппаратная реализация деления.
2) Память:
   - встроенная энергонезависимая Flash-память программ размером 128 Кбайт;
   - встроенное ОЗУ размером 32 Кбайт;
   - контроллер внешней шины с поддержкой микросхем памяти СОЗУ, ПЗУ, NAND Flash.
3) Питание и тактовая частота:
   - внешнее питание 2,2 ÷ 3,6 В;
   - встроенный регулируемый стабилизатор напряжения на 1,8 В для питания ядра;
   - встроенные схемы контроля питания;
   - встроенный домен с батарейным питанием;
   - встроенные подстраиваемые RC генераторы 8 МГц и 40 кГц;
   - внешние кварцевые резонаторы на 2 ÷ 16 МГц и 32 кГц;
   - встроенный умножитель тактовой частоты PLL для ядра;
   - встроенный умножитель тактовой частоты PLL для USB.
4) Режим пониженного энергопотребления:
   - режимы Sleep, Deep Sleep и Standby;
   - батарейный домен с часами реального времени и регистрами аварийного сохранения.
5) Аналоговые модули:
   - два 12-разрядных АЦП (до 16 каналов);
   - температурный датчик;

- двухканальный 12-разрядный ЦАП;
- встроенный компаратор.

6) Периферия:
   - контроллер DMA с функциями передачи Периферия-Память, Память-Память;
   - два контроллера CAN интерфейса;
   - контроллер USB интерфейса с функциями работы Device и Host;
   - контроллеры интерфейсов UART, SPI, I2C;
   - три 16-разрядных таймер-счетчика с функциями ШИМ и регистрации событий;
   - до 96 пользовательских линий ввода-вывода.
7) Отладочные интерфейсы:
   - последовательные интерфейсы SWD и JTAG.
8) Тип корпуса - LQFP64.
9) Ближайший аналог - STM32F103x.
10) Температурный диапазон – минус 45 °C ...+85°C

Таким образом, микроконтроллер может быть использован для решения широкого спектра задач, так как обладает внушительными характеристиками.

На рисунке 1 представлена схема расположения выводов данного микроконтроллера. Стоит заметить, что почти все выводы с портов ввода/вывода имеют альтернативные функции, не представленные на данном рисунке, и могут быть переопределены для выполнения функций различных модулей

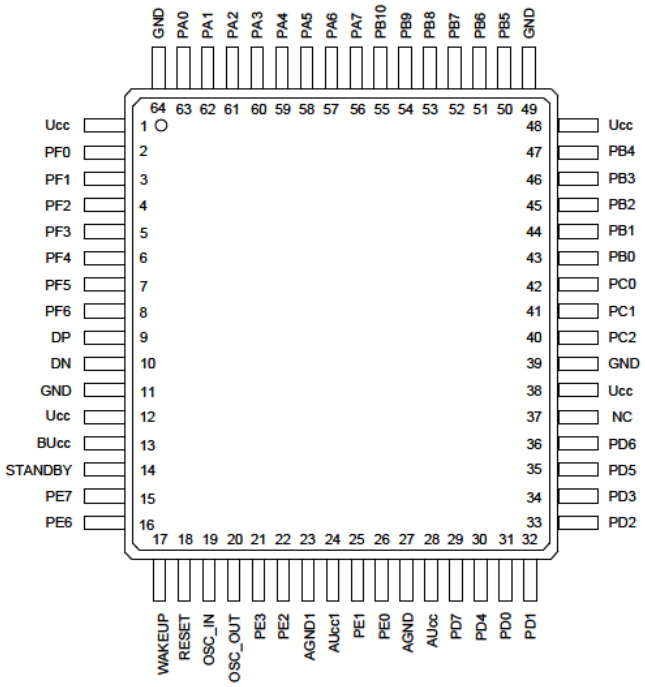

Рисунок 1 – Схема выводов микроконтроллера

Назначение линий портов микроконтроллера приведено в таблице 1. Для того, чтобы линии порта перешли под управление того или иного периферийного блока, необходимо задать для выбранных линий выполняемую функцию и настройки. Обратим внимание на то, что микроконтроллер имеет только 64 вывода, поэтому не каждая линия порта может быть связана с выводом микросхемы.

Таблица 1 - Функции линий портов микроконтроллера К1986ВЕ92QI

Линия	Вывод	Цифровая функция			Аналоговая функция
		Основная	Альтернат.	Переопред.	
1	2	3	4	5	6
Порт A					
PA0	63	DATA0	EXT_INT1	–	–
PA1	62	DATA1	TMR1_CH1	TMR2_CH1	–
PA2	61	DATA2	TMR1_CH1N	TMR2_CH1N	–
PA3	60	DATA3	TMR1_CH2	TMR2_CH2	–
PA4	59	DATA4	TMR1_CH2N	TMR2_CH2N	–
PA5	58	DATA5	TMR1_CH3	TMR2_CH3	–
PA6	57	DATA6	CAN1_TX	UART1_RXD	–
PA7	56	DATA7	CAN1_RX	UART1_TXD	–
Порт B					
PB0	43	DATA16	TMR3_CH1	UART1_TXD	–
PB1	44	DATA17	TMR3_CH1N	UART2_RXD	–
PB2	45	DATA18	TMR3_CH2	CAN1_TX	–
PB3	46	DATA19	TMR3_CH2N	CAN1_RX	–
PB4	47	DATA20	TMR3_BLK	TMR3_ETR	–
PB5	50	DATA21	UART1_TXD	TMR3_CH3	–
PB6	51	DATA22	UART1_RXD	MR3_CH3N	–
PB7	52	DATA23	nSIROUT1	TMR3_CH4	–
PB8	53	DATA24	COMP_OUT	TMR3_CH4N	–
PB9	54	DATA25	nSIRIN1	EXT_INT4	–
PB10	55	DATA26	EXT_INT2	nSIROUT1	–
Порт C					
PC0	42	–	SCL1	SSP2_FSS	–
PC1	41	OE	SDA1	SSP2_CLK	–
PC2	40	WE	TMR3_CH1	SSP2_RXD	–
Порт D					
PD0	31	TMR1_CH1N	UART2_RXD	TMR3_CH1	ADC0_REF+
PD1	32	TMR1_CH1	UART2_TXD	TMR3_CH1N	ADC1_REF-
PD2	33	BUSY1	SSP2_RXD	TMR3_CH2	ADC2
PD3	34	–	SSP2_FSS	TMR3_CH2N	ADC3

Окончание таблицы 1

1	2	3	4	5	6
PD4	30	TMR1_ETR	nSIROUT2	TMR3_BLK	ADC4
PD5	35	CLE	SSP2_CLK	TMR2_ETR	ADC5
PD6	36	ALE	SSP2_TXD	TMR2_BLK	ADC6
PD7	29	TMR1_BLK	nSIRIN2	UART1_RXD	ADC7
Порт E					
PE0	26	ADDR16	TMR2_CH1	CAN1_RX	DAC2_OUT
PE1	25	ADDR17	TMR2_ CH1N	CAN1_TX	DAC2_REF
PE2	22	ADDR18	TMR2_CH3	TMR3_CH1	COMP_IN1
PE3	21	ADDR19	TMR2_CH3N	TMR3_CH1N	COMP_IN2
PE6	16	ADDR22	CAN2_RX	TMR3_CH3	OSC_IN32
PE7	15	ADDR23	CAN2_TX	TMR3_CH3N	OSC_OUT32
Порт F					
PF0	2	ADDR0	SSP1_TXD	UART2_RXD	–
PF1	3	ADDR1	SSP1_CLK	UART2_TXD	–
PF2	4	ADDR2	SSP1_FSS	CAN2_RX	–
PF3	5	ADDR3	SSP1_RXD	CAN2_TX	–
PF4	6	ADDR4	–	–	–
PF5	7	ADDR5	–	–	–
PF6	8	ADDR6	TMR1_CH1	–	–

На рисунке 2 изображена структурная блок-схема микроконтроллера К1986ВЕ92QI, наглядно представляющая наличествующие периферийные устройства и их взаимодействие. Используемые обозначения представлены в таблице 2.

Таблица 2 – таблица обозначений функциональных блоков К1986ВЕ92QI

Блок	**Описание**
Cortex-M3 RISC CORE	Процессорное ядро ARM Cortex-M3 архитектуры RISC
DMA	Контроллер прямого доступа в память
Interrupt	Контроллер прерываний
System timer	Системный таймер
JTAG/SW debug	Отладочный модуль через интерфейс JTAG/SW
AMBA AHB Bus Matrix	Шинная матрица для связи высокоскоростных внутренних компонентов
AHB APB Bridge	Мост для связи с периферией
Flash	Модуль памяти Flash

Окончание таблицы 2

RAM	Модуль памяти RAM
ROM	Модуль памяти ROM
External System Bus	Внешняя системная шина
System Clock Manager	Модуль системного тактирования
UART	Контроллер UART
SPI	Контроллер SPI
BKP Controller	Контроллер резервных данных
Power Detector	Модуль управления питанием
I2C	Контроллер I2C
GPIO	Интерфейс ввода/вывода общего назначения
16 Timer	16-разрядный таймер
ADC Controller	Контроллер аналого-цифрового преобразователя
CAN	Контроллер CAN
DAC Controller	Контроллер цифро-аналогового преобразователя
USB	Контроллер USB
Controller Comparator	Контроллер компаратора
WDT Controller	Контроллер сторожевого таймера
CPU PLL	Фазовая автоподстройка частоты для процессорного ядра
USB PLL	Фазовая автоподстройка частоты для USB
HSI	Высокоскоростной внутренний генератор тактовой частоты
LSI	Низкоскоростной внутренний генератор тактовой частоты
HSE	Высокоскоростной внешний генератор тактовой частоты
LSE	Низкоскоростной внешний генератор тактовой частоты
Real Time Clock BKP memory	Резервная память
LDO Cap Less	Регулятор напряжения
ADC	АЦП
DAC	ЦАП
PSY USB	Дескриптор USB
Comparator	Компаратор
IWDT	Независимый сторожевой таймер

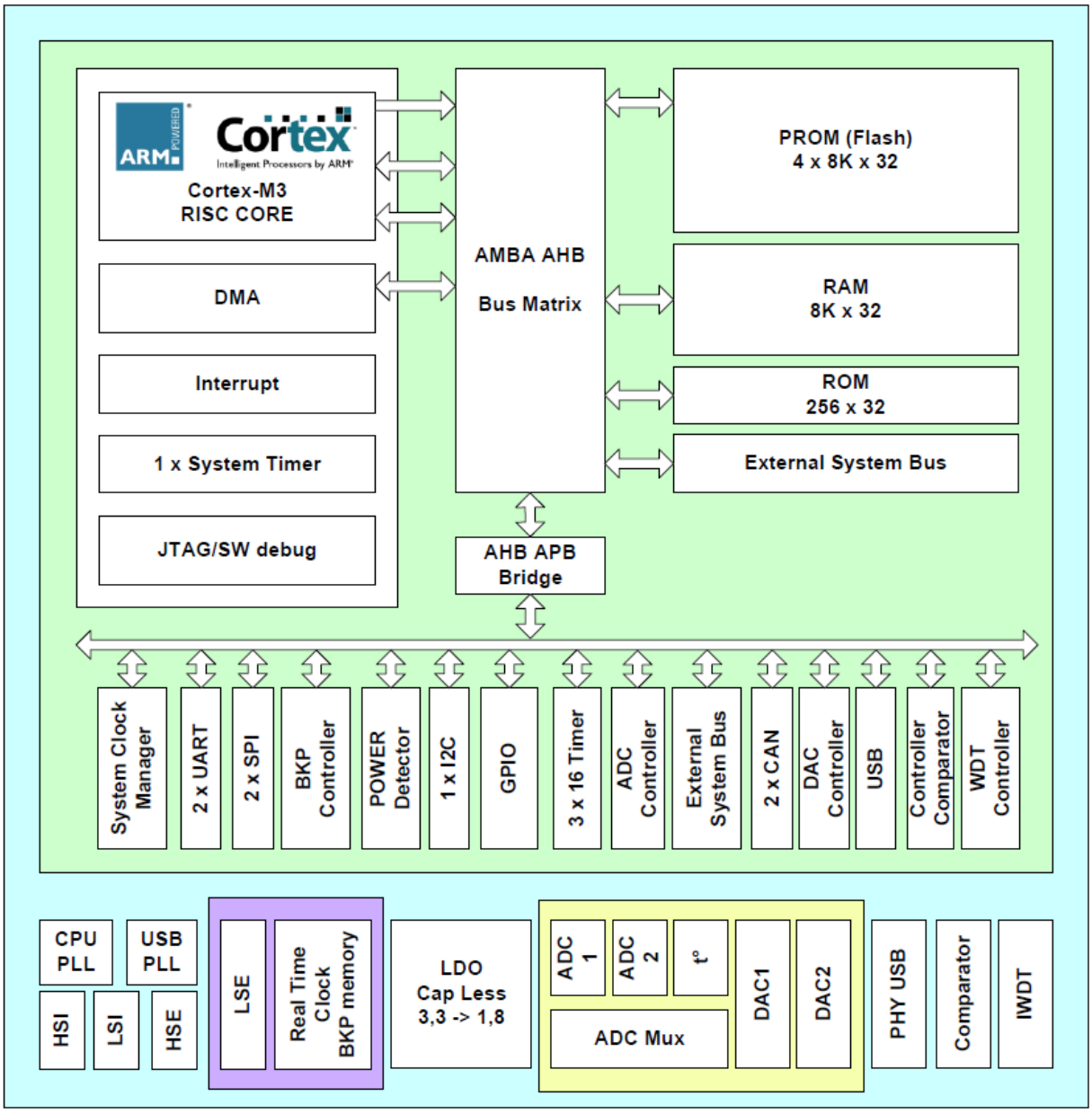

Рисунок 2 – Структурная блок-схема микроконтроллера К1986ВЕ92QI

Процессорное ядро имеет три системных шины:

- I Code – шина выборки инструкций;
- D Code – шина выборки данных, расположенных в коде программы;
- S Bus – шина выборки данных, расположенных в области ОЗУ.

Также в микроконтроллере реализован контроллер прямого доступа в память (DMA), который осуществляет выборку через шину DMA Bus. Структурная схема организации памяти представлена на рисунке 3.

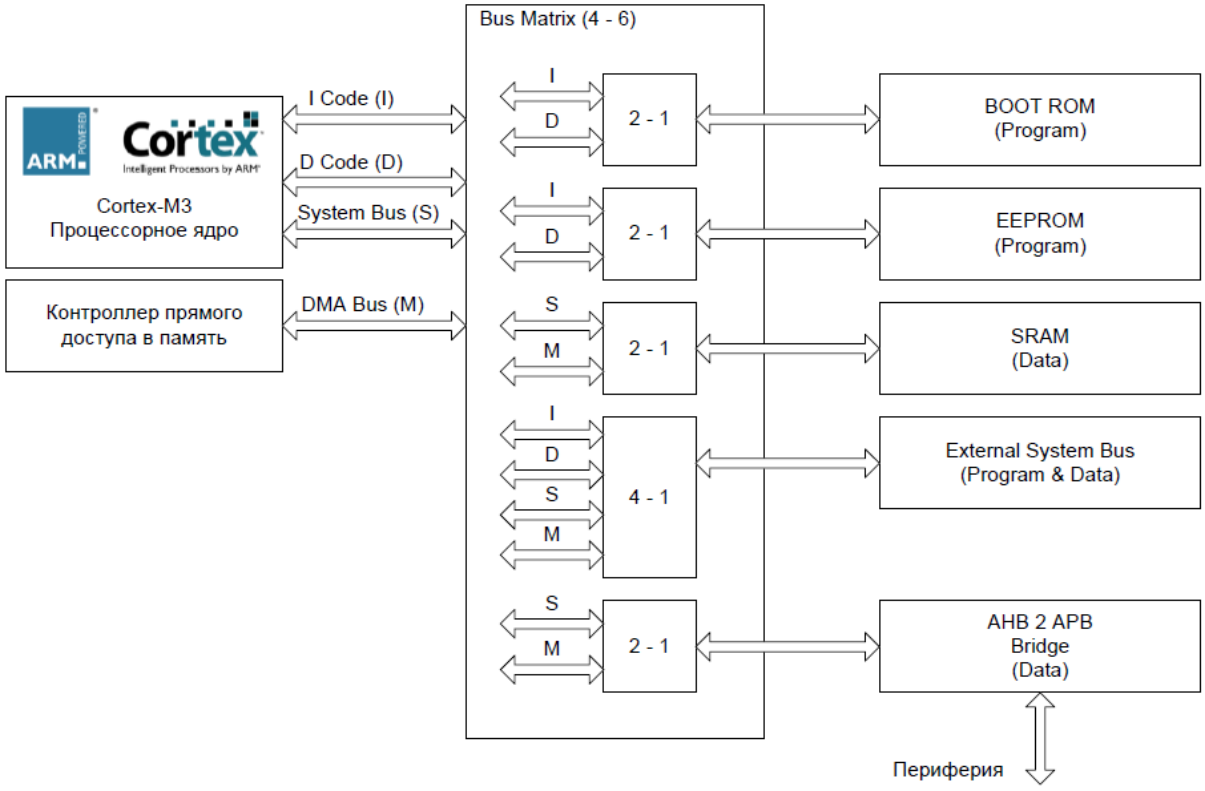

Рисунок 3 – Структурная схема организации памяти

Все адресное пространство микроконтроллера едино и имеет максимальный объем 4 Гбайт. В данное адресное пространство отображаются различные модули памяти и периферии.

По умолчанию для записи программ используется область памяти 0x08000000 - 0x0801FFFFF внутренней Flash памяти.

После включения питания и снятия внутренних (POR) и внешних (RESET) сигналов сброса, микроконтроллер начинает выполнять программу из загрузочной области ПЗУ BOOT ROM. В загрузочной программе микроконтроллер определяет, в каком из режимов он будет функционировать, и переходит в этот режим. Режим функционирования определяется внешними выводами MODE[2:0], при этом перед опросом состояния этих выводов, для них включается внутренняя подтяжка к шине «Общий» (встроенные резисторы подтяжки к шине «Общий» имеют сопротивление ~50 кОм). Также устанавливается бит FPOR в регистре BKP_REG_0E, который может быть сброшен только при отключении основного питания UCC. После перезапуска микроконтроллера уровни на выводах MODE[2:0] не влияют на режим функционирование микроконтроллера, если установлен бит FPOR.

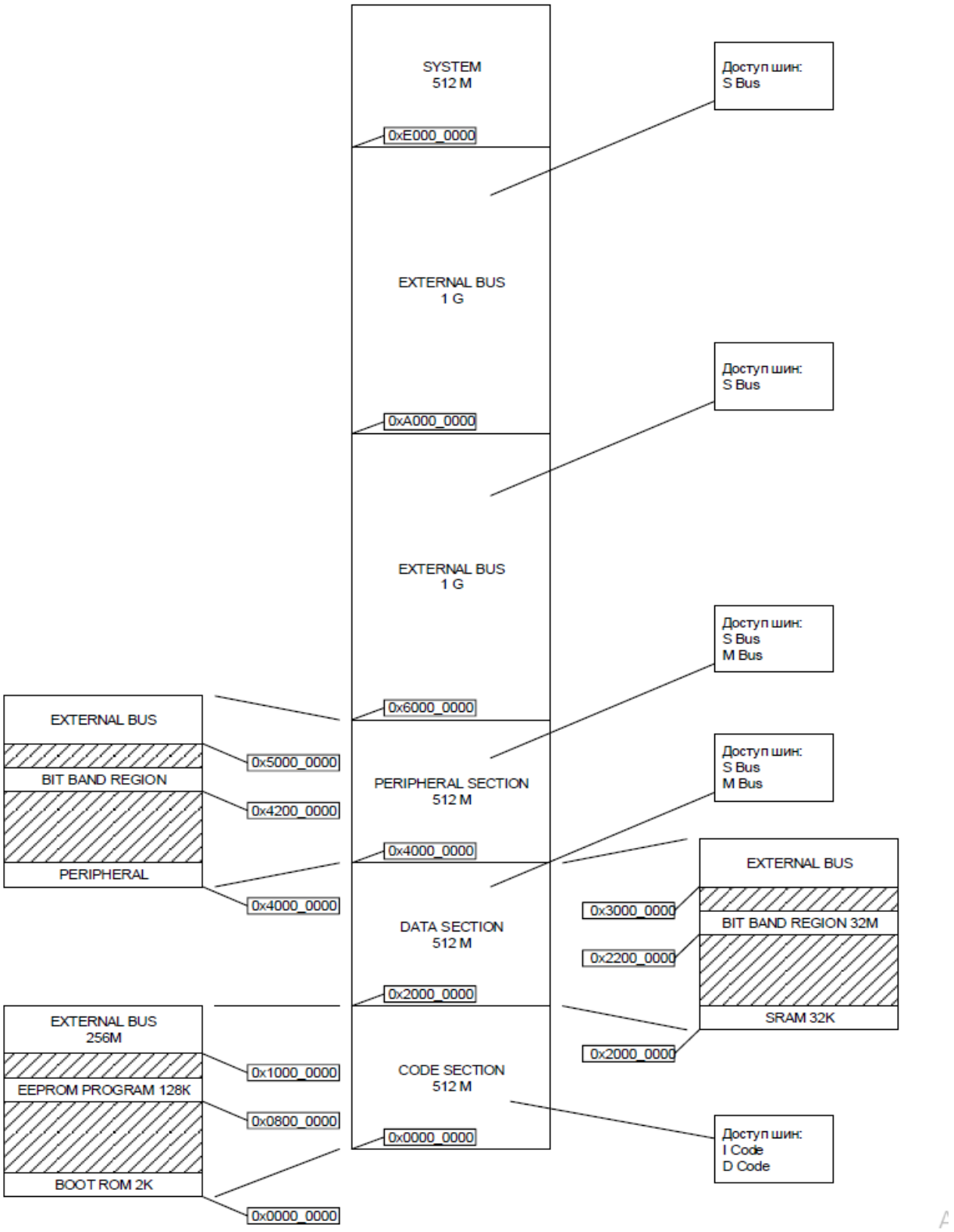

Рисунок 4 - Карта распределения основных областей памяти

При работе с отладочной платой MODE[2:0] устанавливается с помощью переключателей на плате. Режимы запуска МК для JTAG представлены в таблице 3 .

Таблица 3 – Режимы первоначального запуска микроконтроллера

MODE [2:0]	Стартовый адрес	Описание	Порты	Описание выводов интерфейса
000	0x0800_0000	Процессор начинает выполнять программу из внутренней Flash-памяти программ. При этом установлен отладочный интерфейс JTAG_B	PD2/JB_TRST PD1/JB_TCK PD0/JB_TMS PD3/JB_TDI PD4/JB_TDO	В качестве выводов интерфейса используются выводы порта D, совмещенные с каналами АЦП, выводами каналов Таймера 1 и 3, UART2 и SSP2, использование которых при отладке запрещено
001	0x0800_0000	Процессор начинает выполнять программу из внутренней Flash-памяти программ. При этом разрешается работа отладочного интерфейса JTAG_A	PB4/JA_TRST PB2/JA_TCK PB1/JA_TMS PB3/JA_TDI PB0/JA_TDO	В качестве выводов интерфейса используются выводы порта B, совмещенные с выводами данных внешней системной шины, выводами таймера 3, выводами UART1 и UART2 и CAN1, использование которых при отладке запрещено

1.2 Описание интерфейсов и контроллеров интерфейсов

Особый интерес в данной работе представляют внешние интерфейсы работы, так как именно они будут использоваться для передачи информации между двумя отладочными платами. В данном микроконтроллере представлено 5 интерфейсных модулей:

- Контроллер интерфейса MDR_USB
- Контроллер интерфейса MDR_CAN
- Контроллер интерфейса MDR_I2C
- Контроллер MDR_SSP
- Контроллер MDR_UART

Рассмотрим каждый из модулей более детально с описанием интерфейса в целом.

1.2.1 USB

1.2.1.1 Описание интерфейса

USB – последовательный интерфейс, используемый для подключения периферийных устройств. Соответственно, существуют понятие «главное устройство» (хост, он управляет обменом данными через интерфейс, выступает инициатором обмена) и «периферийное устройство» (клиент, в процессе обмена данными «подчиняется» хосту).

Логика работы у хоста и клиента принципиально отличается, соответственно нельзя напрямую соединять устройства «хост – хост» и «клиент – клиент».

Физически интерфейс USB использует 4 провода: «земля (GND)», «+5В (VBUS)», «D+», «D-». Первые два могут использоваться для питания периферийного устройства (максимальный ток 500 мА). Два последних служат для передачи данных.

На логическом уровне, обмен данными происходит через некоторые логические, виртуальные каналы внутри одного физического USB интерфейса. Такие каналы называют «Конечными точками» (EndPoints).

Конечные точки (каналы) бывают 4 видов:

Control – данный тип канала используется хостом для управления периферийным устройством. Хотя иногда данный тип канала используется для передачи данных.

Bulk — данный тип канала используется для обмена данными. Гарантирование целостности данных и гарантированная доставка данных для данного типа канала реализована «в железе». Однако скорость передачи данных по такому каналу ограничена.

Isochronous — данный тип канала в основном используется для обмена потоковыми данными. Целостность и доставка данных не контролируются, зато скорость значительно выше чем для Bulk каналов.

Interrupt – используются для реализации подобия «прерываний». Такие «прерывания» являются логическими, и никак напрямую не связанны с аппаратными прерываниями МК или прерываниями ОС.

Минимальная реализация USB устройства требует наличие всего одного Control канала (так называемая «нулевая конечная точка»). Остальные типы каналов, как и их количество определяет разработчик устройства исходя из функций устройства.

1.2.1.2 Общее описание контроллера интерфейса

Контролер USB реализует функции контроллера функционального устройства (Device) и управляющего устройства (Host) в соответствии со спецификацией USB 2.0.

Контроллер USB поддерживает следующие возможности: режимы работы Full Speed (12 Мбит/с) и Low Speed (1.5 Мбит/с), контроль ошибок с помощью циклического избыточного кода (CRC), NRZI код приема/передачи, управляющая (Control), сплошная (Bulk), изохронная (Isochronous) передачи и передача по прерываниям (Interrupt), конфигурирование USB Device от 1-й до 4-х оконечных точек; каждая оконечная точка USB Device имеет собственную память FIFO размером 64 байта. USB Host поддерживает до 16 оконечных точек. Возможности USB Host: FIFO размером 64 байта; автоматическая отправка SOF пакетов; вычисление оставшегося во фрейме времени.

Контроллер USB может быть сконфигурирован как USB Host или как USB Device. Конфигурация задается битом HOST_MODE в регистре HSCR (0 – режим Device, 1 – режим Host). Прием/передача через физический интерфейс USB разрешаются установкой бит EN_RX и EN_TX в этом же регистре. В режиме приема имеется возможность отключить передатчик в целях экономии потребления (EN_TX=0). Отключение всего блока в целом осуществляется при EN_RX=0.

В режиме Device параметры шины задаются в регистре SC. Скорость задается битом SCFSR (0 – 1,5 Мбит/с, 1 – 12 Мбит/с), полярность битом SCFSP (0 – Low speed, 1 – Full speed) этого регистра.

В режиме Host параметры шины задаются в регистре HTXLC. Скорость задается битом FSLR (0 – 1,5 Мбит/с, 1 – 12 Мбит/с), полярность битом FSPL (0 – Low speed, 1 – Full speed) этого регистра.

В режиме Host контроллер автоматически определяет подключение или отключение устройства к шине. Бит CONEV регистра USB_HSI устанавливается в 1 при возникновении одного из событий.

1.2.1.3 Задание адреса и инициализация оконечных точек

Функциональный адрес устройства USB задается в регистре SA.

Для инициализации конечной точки в первую очередь необходимо установить бит глобального разрешения всех оконечных точек (SCGEN = 1 в регистре SC). Биты EPEN в регистрах SEP[x].CTRL должны быть установлены, чтобы разрешить соответствующую оконечную точку. Если предполагается использовать изохронный тип передачи оконечной точки, то необходимо установить бит EPISOEN в соответствующем регистре SEP[x].CTRL.

1.2.2 CAN

1.2.2.1 Описание интерфейса

CAN (англ. Controller Area Network — сеть контроллеров) — стандарт промышленной сети, ориентированный, прежде всего, на объединение в единую сеть различных исполнительных устройств и датчиков. Режим передачи — последовательный, широковещательный, пакетный. CAN разработан компанией Robert Bosch GmbH в середине 1980-х и в настоящее время широко распространён в промышленной автоматизации, технологиях «умного дома», автомобильной промышленности и многих других областях. Стандарт для автомобильной автоматики.

Информация на шине представлена в виде фиксированных сообщений различной, но ограниченной длины. Когда шина свободна, любой подключенный узел может начать передавать новое сообщение. При передаче информации с помощью протокола CAN используется четыре типа пакетов:

- пакет данных (data frame) — передаёт данные, является самым часто используемым пакетом;
- пакет удаленного запроса (remote frame) — служит для запроса на передачу кадра данных с тем же идентификатором;
- пакет перегрузки (overload frame) — обеспечивает промежуток между кадрами данных или запроса;
- пакет ошибки (error frame) — передаётся узлом, обнаружившим в сети ошибку.

Пакет данных состоит из 7 различных полей:

- "начало пакета" (SOF-start of frame);
- "поле арбитража" (arbitration field);
- "поле контроля" (control field);
- "поле данных" (data field);
- "поле CRC" (CRC field);
- "поле подтверждения" (ACK field);
- "конец пакета" (end of frame).

Поле данных может иметь нулевую длину.

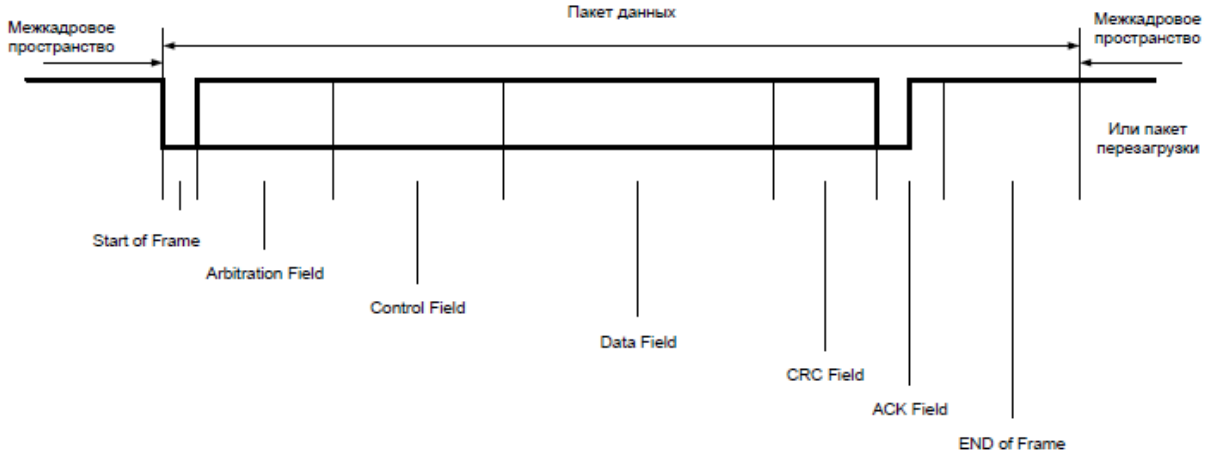

Рисунок 5 – Пакет сообщения CAN

В терминах протокола CAN логическая единица называется рецессивным битом, а логический ноль называется доминантным битом. Во всех случаях доминантный бит будет подавлять рецессивный. То есть, если несколько узлов выставят на шину рецессивный бит, а один – доминантный, то обратно всеми узлами будет считан доминантный бит.

При свободной шине любой узел может начинать передачу в любой момент. В случае одновременной передачи кадров двумя и более узлами проходит арбитраж доступа: передавая идентификатор, узел одновременно проверяет состояние шины. Если при

передаче рецессивного бита принимается доминантный — считается, что другой узел передаёт сообщение с большим приоритетом, и передача откладывается до освобождения шины. Таким образом, в отличие, например, от Ethernet в CAN не происходит непроизводительной потери пропускной способности канала при коллизиях. Цена этого решения — возможность того, что сообщения с низким приоритетом никогда не будут переданы.

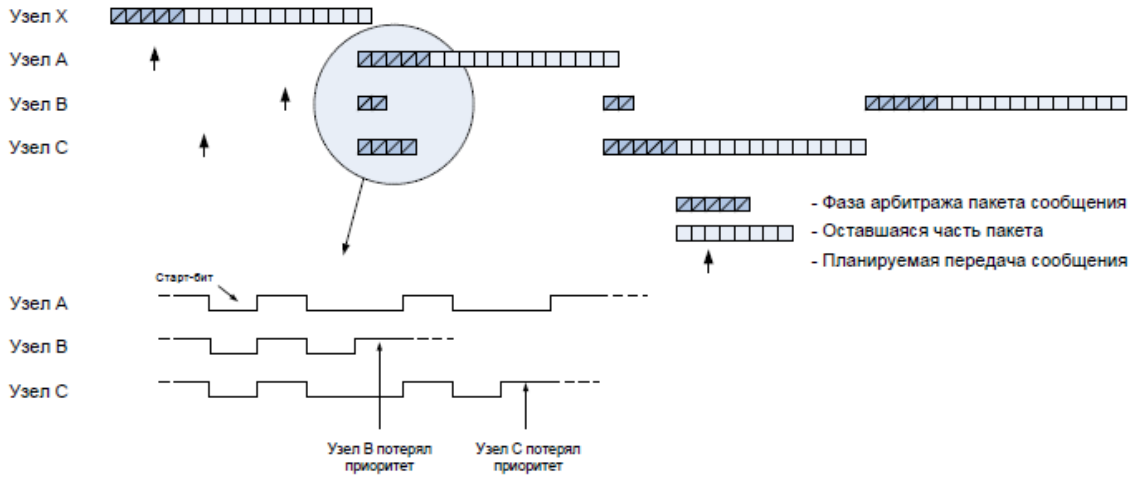

Рисунок 6 – Арбитраж на шине CAN

1.2.2.2 Общее описание контроллера интерфейса

В микроконтроллере реализовано два независимых контроллера интерфейса CAN. Они являются полнофункциональными CAN-узлами, отвечающими требованиям к активным и пассивным устройствам CAN 2.0A и 2.0B и поддерживающими передачу данных на скорости не более 1 Мбит/сек.

Интерфейс CAN позволяет обмениваться сообщениями в сети равноправных устройств. При передаче сообщения в сети CAN все узлы сети получают это сообщение. В сообщении передается уникальный идентификатор и данные. Все сообщения в протоколе CAN могут содержать не более восьми байтов данных. При возникновении коллизий (одновременная передача сообщений различными узлами) при передаче идентификатора происходит арбитраж, и узел передающий сообщение с большим номером идентификатора уступает сеть узлу передающему сообщение с меньшим номером идентификатора.

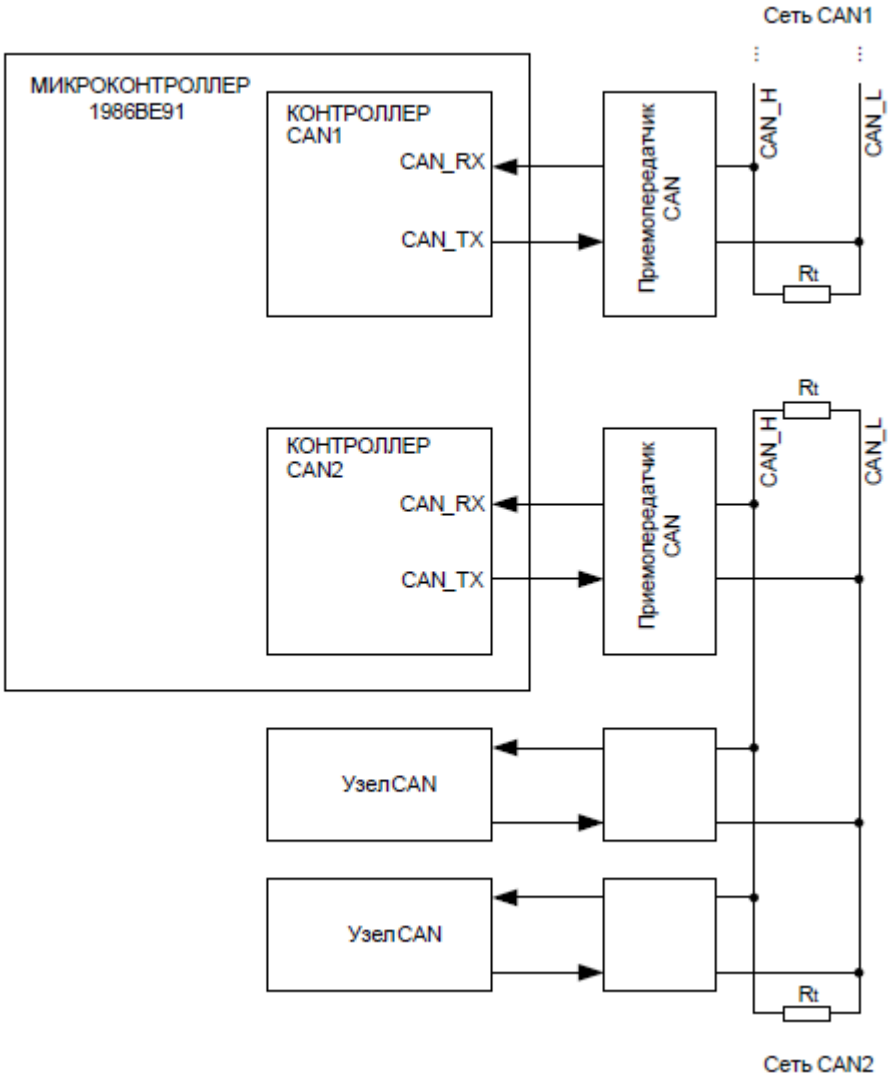

Рисунок 7 - Структурная схема организации сети CAN

Особенности:

- поддержка CAN протокола версии CAN 2.0 A и B;
- скорость передачи до 1 Мбит/с;
- 32 буфера приема/передачи;
- поддержка приоритетов сообщений;
- 32 фильтра приема;
- маскирование прерываний.

1.2.2.3 Прием и передача сообщений

На рисунке 8 представлена структурная блок-схема контроллера CAN

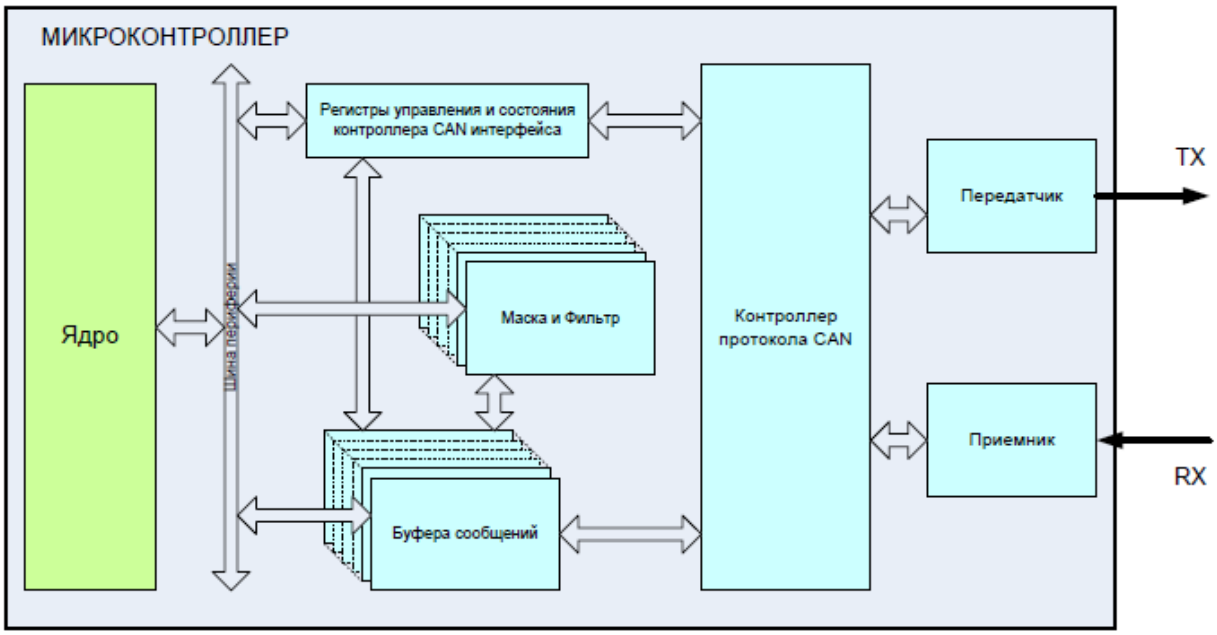

Рисунок 8 - Структурная блок-схема контроллера CAN

Для передачи сообщения необходимо в разрешенный для работы и конфигурируемый на передачу буфер записать сообщение для передачи (задать значения регистрам CAN_BUF[x].ID, CAN_BUF[x].DLC, CAN_BUF[x].DATAL и CAN_BUF[x].DATAH), после чего установить бит TX_REQ. После установки этого бита сообщение будет поставлено в очередь на отправку. После отправки сообщения бит TX_REQ будет автоматически сброшен. Если в нескольких буферах есть сообщения на отправку, то порядок отправки определяется по полю PRIOR_0. Если у сообщения бит PRIOR_0 выставлен в ноль, то оно отправляется в первую очередь. Если есть несколько сообщений с одинаковым приоритетом, то порядок отправки определяется порядковым номером буфера, буфер с меньшим порядковым номером имеет больший приоритет. Значение полей ID для выбора порядка отправки в рамках контроллера CAN (одного узла) значения не имеет. По ID выбирается приоритет между различными узлами.

Для приема сообщений необходимо иметь свободные и разрешенные для работы буфера, сконфигурированные на прием сообщений. При этом если по шине CAN будут передаваться сообщения от других узлов, они будут сохраняться в этих буферах.

1.2.3 Описание модуля SSP

1.2.4 Описание модуля UART

1.2.5 Описание модуля I2C

1.3 Описание отладочной платы

1.3.1 здесь надо будет расписать про все модули

1.4 Алгоритмы работы программы