**Блок инициализации порта**

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

Блок реализует настройку порта микропроцессора. Программная реализация блока, расположена в заголовочном файле port.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\_port().

Всего у микросхемы K1986ВЕ92QI, помещенной в корпус LQFP64,

есть 43 линии ввода-вывода, объединенные в 6 портов, как показано в таблице 2.2.

Порты и линии ввода-вывода микроконтроллера K1986ВЕ92QI

Наименование порта Количество линий Наименование линий

PORTA 8 PA0…PA7

PORTB 11 PB0…PB10

PORTC 3 PC0…PC2

PORTD 8 PD0…PD7

PORTE 6 PE0…PE3, PE6…PE7

PORTF 7 PF0…PF6

Файл port.h :

void Init\_port(uint16\_t Pin, uint16\_t PORT\_MODE, uint16\_t PORT\_FUNC, uint16\_t PORT\_OE,

uint16\_t PORT\_SPEED, MDR\_PORT\_TypeDef\* MDR\_PORT)

{

// Структура для инициализации портов

PORT\_InitTypeDef PortInitStructure;

// Разрешить тактирование требуемых периферийных устройств

RST\_CLK\_PCLKcmd ( PCLK\_BIT(MDR\_PORT), ENABLE);

// Сделать выходами порты, к которым подключены светодиоды

PORT\_StructInit (&PortInitStructure);

PortInitStructure.PORT\_Pin = Pin;

PortInitStructure.PORT\_MODE = PORT\_MODE;

PortInitStructure.PORT\_FUNC = PORT\_FUNC;

PortInitStructure.PORT\_OE = PORT\_OE;

PortInitStructure.PORT\_SPEED = PORT\_SPEED;

PORT\_Init (MDR\_PORT, &PortInitStructure);

}

**Входные порты**

**нет**

**Выходные порты**

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

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера пинов** | **Pins** | линии выбранного порта их может быть произвольное количество  они формируются по схеме ИЛИ. | Pin\_0 |
| **Тип порта** | **PORT\_MODE** | указывают режим работы выбранных пинов.  Возможны два варианта значений:  PORT\_MODE\_ANALOG – линия является аналоговой;  PORT\_MODE\_DIGITAL – линия является цифровой.  Режим аналоговой линии выбирается, если требуется работать с АЦП, ЦАП, аналоговым компаратором или внешним низкочастотным кварцевым резонатором. В остальных случаях используется режим цифровой линии. | PORT\_MODE\_DIGITAL |
| **Функция** | **PORT\_FUNC** | конфигурация линий, если линия сконфигурирована как цифровая, то ее нужно дополнительно сконфигурировать, выбрав одну из возможных функций.  Для этого в поле PORT\_FUNC типа PORT\_FUNC\_TypeDef заносят одно из следующих значений:  PORT\_FUNC\_PORT – линия используется, как цифровой вход или выход;  PORT\_FUNC\_MAIN – для линии используется основная функция;  PORT\_FUNC\_ALTER – для линии используется альтернативная функция;  PORT\_FUNC\_OVERRID – для линии используется перегруженная функция. | PORT\_FUNC\_PORT |
| **Направление ввода** | **PORT\_OE** | Вход или выход. Если требуется сделать пин цифровым входом, то напишем:  PortInitStructure.PORT\_OE = PORT\_OE\_IN;  А если цифровым выходом, то:  PortInitStructure.PORT\_OE = PORT\_OE\_OUT; | PORT\_OE\_OUT |
| **Скорость** | **PORT\_SPEED** | Показывают скорость работы линии, т.е. какой частоты сигнал может через нее проходить.  Возможны следующие значения:  PORT\_OUTPUT\_OFF – выход выключен;  PORT\_SPEED\_SLOW – низкая скорость (фронт порядка 100 нс, частота до 5 МГц);  PORT\_SPEED\_FAST – высокая скорость (фронт порядка 20 нс, частота до 25 МГц);  PORT\_SPEED\_MAXFAST – предельно высокая скорость (фронт порядка 10 нс, частота до 50 МГц). | PORT\_SPEED\_FAST |
| **Порт** | **MDR\_PORT** | указатель на порт | MDR\_PORTA\_BASE |

**Пример port.prt**

**Блок PORT\_WriteBit записи значений (0 или 1) в выбранные линии**

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

Блок использует функцию PORT\_WriteBit() из библиотеки MDR32F9Qx\_StdPeriph\_Driver и позволяет установить выбранные выходы в указанное состояние (Bit\_ RESET – в 0; Bit\_SET – в 1). Если необходимо оптимизировать работу функции, то ее можно заменит используя прямое обращение к регистрам микропроцессора.

**Входные порты**

Pins\_in значение (0 или 1) на выбранных пинах;

**Выходные порты**

Ready алгоритм записи выполнен

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера пинов** | **Pins** | линии выбранного порта их может быть произвольное количество  они формируются по схеме ИЛИ. | Pin\_0 |
| **Порт** | **PORT** | Указатель на порт | MDR\_PORTA\_BASE |

**Пример port.prt**

**Блок PORT\_Write записи значений в выбранные линии порта**

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

Блок использует функцию PORT\_Write() из библиотеки MDR32F9Qx\_StdPeriph\_Driver и позволяет установить выбранные выходы в указанное состояние (часть выходов может иметь значение 0, а другая 1), в зависимости от вектора входных значений. Если необходимо оптимизировать работу функции, то ее можно заменит используя прямое обращение к регистрам микропроцессора.

**Входные порты**

Pins\_in массив значений для выбранных в свойствах блока пинов пинов, для каждого пина возможно два значения либо 0 или 1;

Pins\_out массив значений определенный для каждой линии связи

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера пинов** | **Pins** | линии выбранного порта их может быть произвольное количество  они формируются по схеме ИЛИ. | Pin\_0 |
| **Порт** | **PORT** | Указатель на порт | MDR\_PORTA\_BASE |
| **Размер массива входных данных** | **Size** | Размер входного массива данных, равен количеству выбранных линий порта |  |
|  |  |  |  |

**Пример port.prt**

**Блок PORT\_SetResetBits записи (0 | 1) в выбранные линии порта**

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

Блок использует две функции PORT\_SetBits() и PORT\_ResetBits() из библиотеки MDR32F9Qx\_StdPeriph\_Driver и позволяет установить выбранные линии порта одновременно в одно из двух положений (1|0).

**Входные порты**

set\_bit может принимать одно из двух значений (1|0)

**Выходные порты**

Set значение установленное на линиях связи (1|0)

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера пинов** | **Pins** | линии выбранного порта их может быть произвольное количество  они формируются по схеме ИЛИ. | Pin\_0 |
| **Порт** | **PORT** | Указатель на порт | MDR\_PORTA\_BASE |

**Пример port.prt**

**Блок** PORT\_ReadInputData **читает состояния всех входов указанного порта, возвращая 16-битное целое число**

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

Блок использует две функции PORT\_ReadInputData () из библиотеки MDR32F9Qx\_StdPeriph\_Driver. Функция PORT\_ReadInputData() читает состояние всех входов указанного порта, возвращая 16-битное целое число (по количеству линий в порте):

uint16\_t in\_data;

...

in\_data = PORT\_ReadInputData (MDR\_PORTB);

**Входные порты**

Read\_data определяет последовательность включения блока в работу

**Выходные порты**

Data\_port значение установленное на линиях связи в виде 16 битового целого числа (по количеству линий в порте)

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Порт** | **PORT** | Указатель на порт | MDR\_PORTA\_BASE |
|  |  |  |  |

**Пример port.prt**

**Блок** PORT\_ReadInputDataBit **читает состояние определенного**

**входа.**

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

Блок использует две функции PORT\_ReadInputData () из библиотеки MDR32F9Qx\_StdPeriph\_Driver. PORT\_ReadInputDataBit () читает состояние определенного входа. В следующем примере производится чтение состояния входа PB0 .

ui nt 8\_ t flag ;

...

flag = PORT\_ReadInputDataBit (MDR\_PORTB, PORT\_Pin\_0);

**Входные порты**

Read\_data определяет последовательность включения блока в работу

**Выходные порты**

Data\_bit значение установленное на выбранной линии порта

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера пинов** | **Pins** | линии выбранного порта их может быть произвольное количество  они формируются по схеме ИЛИ. | Pin\_0 |
| **Порт** | **PORT** | Указатель на порт | MDR\_PORTA\_BASE |

**Пример port.prt**

**Блок PORT\_WriteBits записи (0 | 1) в выбранные линии порта**

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

Блок использует одна функция PORT\_WriteBit () из библиотеки MDR32F9Qx\_StdPeriph\_Driver и позволяет установить выбранные линии порта одновременно в одно из двух положений (1|0).

**Входные порты**

set\_bit может принимать одно из двух значений (1|0)

**Выходные порты**

Data\_bits значение установленное на линиях связи (1|0)

**Свойства**

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Имя | Описание | По умолчанию |
| **Номера пинов** | **Pins** | линии выбранного порта их может быть произвольное количество  они формируются по схеме ИЛИ. | Pin\_0 |
| **Порт** | **PORT** | Указатель на порт | MDR\_PORTA\_BASE |

**Пример port.prt**