

# Микроконтроллер MDR32F9Q21

## Часть 2. Работа с портами микроконтроллера

Михаил Голубцов (Москва)

Цикл статей предназначен для тех читателей, которые хотят ознакомиться с отечественным микроконтроллером, совместимым с Cortex-M3, и научиться его использовать. Рассмотрены основные аппаратные узлы микроконтроллера, а также их программирование. В каждой статье приводится пример реального использования, со схемой и полным текстом программы. Поскольку микроконтроллер Cortex-M3 сложнее многих популярных приборов, данная публикация может облегчить его освоение.

Микроконтроллер содержит шесть 16-разрядных портов ввода-вывода с названиями PORTA – PORTF. Каждый вывод порта может выполнять одну из четырёх возможных функций:

- порт;
- основная функция;
- альтернативная функция;
- переопределённая функция.

Для выбора функции предусмотрено переключение вывода на различные функциональные блоки внутри микроконтроллера. Каждый вывод порта можно настраивать независимо от настроек других выводов. На рисунке 1 показана структура одного вывода порта микроконтроллера.

При работе в режиме порта ввода-вывода имеются дополнительные возможности. В режиме работы порта на выход его можно сконфигурировать как управляемый драйвер или открытый исток, а также установить параметр мощности, который задаёт медленный, средний или короткий фронт на выводе порта при переключении его состояния.

В режиме работы порта на вход можно задать:

- аналоговый или цифровой режим работы;
- триггер Шмитта;
- фильтр на входе;
- подтяжку к питанию;

- подтяжку к общему проводу.

В таблице приведены варианты возможных функций для каждого вывода порта микроконтроллера. Управление портами микроконтроллера осуществляется при помощи набора из восьми 32-разрядных регистров:

MDR_PORTx->RXTX	Данные порта
MDR_PORTx->OE	Направление для каждого вывода порта
MDR_PORTx->FUNC	Режим работы порта (функция)
MDR_PORTx->ANALOG	Аналоговый режим работы порта
MDR_PORTx->PULL	Подтяжки к питанию или общему проводу
MDR_PORTx->PD	Режим работы выходного драйвера
MDR_PORTx->PWR	Мощность передатчика
MDR_PORTx->GFEN	Входной фильтр

Ниже приведено подробное описание регистров.

### MDR\_PORTx->RXTX

Регистр предназначен для управления состоянием выходных линий порта, а также для получения состояния входных линий порта. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Биты с 0 по 15 соответствуют выводам порта соответственно с Rx0 по Rx15. Для установления состояния выходной линии порта в соответствующий ей разряд регистра надо записать «1». Для получения состояния линии порта, настроенной на вход, следует считать из этого регистра число и в разряде, соответствующем линии порта, получить её состояние.

### MDR\_PORTx->OE

Регистр предназначен для задания направления передачи данных на выводах порта. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Биты с 0 по 15 соответствуют выво-

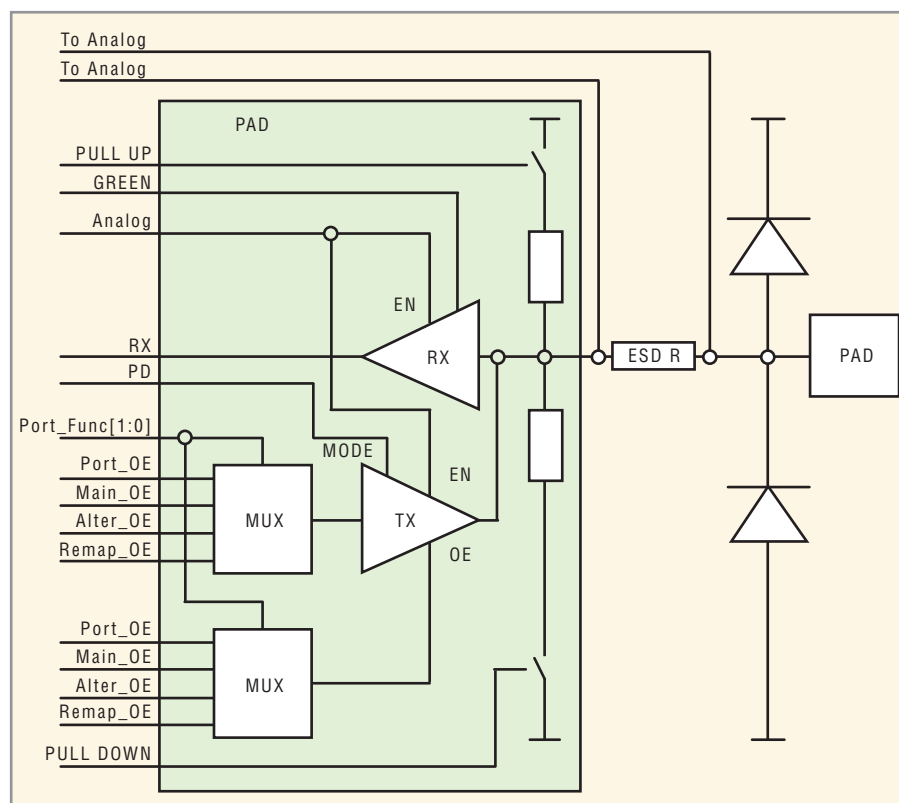


Рис. 1. Структура вывода порта микроконтроллера

дам порта соответственно с Rx0 по Rx15. Для установления направления работы линии порта следует в соответствующий ей разряд регистра записать «0» или «1». При значении «0» линия порта работает на вход, при «1» – на выход.

#### MDR\_PORTx→ANALOG

Регистр предназначен для задания аналогового или цифрового режима работы линий порта. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Биты с 0 по 15 соответствуют выводам порта соответственно с Rx0 по Rx15. Для установления режима работы линии порта следует в соответствующий ей разряд регистра записать «0» или «1». При значении «0» линия порта работает в аналоговом режиме, при «1» – в цифровом режиме.

#### MDR\_PORTx→GFEN

Регистр предназначен для включения или отключения фильтра на линии порта при выбранном для неё направлении работы на вход. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Биты с 0 по 15 соответствуют выводам порта соответственно с Rx0 по Rx15. Для включения или отключения фильтра следует в соответствующий линии порта разряд регистра записать «0» или «1». При значении «0» фильтр выключен, при «1» – фильтр включен.

#### MDR\_PORTx→FUNC

Регистр предназначен для задания режима работы линии порта. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Каждой линии порта соответствует пара из двух битов регистра. Регистр можно представить в виде:

Разряд регистра	31...1	0
Назначение	MODE15[1:0]...	MODE1[1:0]MODE0[1:0]

Для выбора режима работы следует в соответствующую данной линии порта пару битов записать значение:

- «00» – порт ввода/вывода;
- «01» – основная функция;
- «10» – альтернативная функция;
- «11» – переопределённая функция.

#### MDR\_PORTx→PWR

Регистр предназначен для задания выходной мощности линии порта, настроенного для работы на выход. Чем больше мощность, тем короче фронт импульса на линии при изменении её состояния. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Каждой линии порта соответствует пара из двух битов регистра. Регистр можно представить в виде:

Разряд регистра	31...1	0
Назначение	PWR15[1:0] ...	PWR1[1:0]PWR0[1:0]

Для выбора требуемой мощности следует в соответствующую данной линии порта пару битов записать значение:

- «00» – зарезервировано (передатчик отключен);
- «01» – медленный фронт;
- «10» – быстрый фронт;
- «11» – максимально быстрый фронт.

Следует учитывать, что при переключении режима мощности изменяется потребляемая микроконтроллером мощность.

#### MDR\_PORTx→PULL

Регистр предназначен для подключения или отключения встроенных резисторов подтяжки на выводах порта, настроенных для работы на вход. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. В регистре есть две группы логически разделённых битов. Биты с 0 по 15 позволяют управлять резистором подтяжки к питанию. Бит 0 соответствует линии Rx0, бит 1 соответствует линии Rx1, и так далее до 15 бита.

Биты с 16 по 31 позволяют управлять резистором подтяжки к общему проводу (нулю). Бит 16 соответствует линии Rx0, бит 17 соответствует линии Rx1, и так далее до 31 бита. Для задания состояния резистора подтяжки следует в соответствующий данной линии порта разряд регистра записать «0» или «1». При значении «0» подтяжка отключена, при «1» – подтяжка включена.

Функции портов микроконтроллера

Вывод	Функция			
	порт	основная	альтернативная	переопределенная
<b>Порт А</b>				
PA0	PA0	DATA0	EXT_INT1	-
PA1	PA1	DATA1	TMR1_CH1	TMR2_CH1
PA2	PA2	DATA2	TMR1_CH1N	TMR2_CH1N
PA3	PA3	DATA3	TMR1_CH2	TMR2_CH2
PA4	PA4	DATA4	TMR1_CH2N	TMR1_CH2N
PA5	PA5	DATA5	TMR1_CH3	TMR2_CH3
PA6	PA6	DATA6	CAN1_TX	UART1_RXD
PA7	PA7	DATA7	CAN1_RX	UART1_TXD
<b>Порт В</b>				
PB0	PB0	DATA16	TMR3_CH1	UART1_TXD
PB1	PB1	DATA17	TMR3_CH1N	UART2_RXD
PB2	PB2	DATA18	TMR3_CH2	CAN1_TX
PB3	PB3	DATA19	TMR3_CH2N	CAN1_RX
PB4	PB4	DATA20	TMR3_BLK	TMR3_ETR
PB5	PB5	DATA21	UART1_TXD	TMR3_CH3
PB6	PB6	DATA22	UART1_RXD	TMR3_CH3N
PB7	PB7	DATA23	nSIRIOUT1	TMR3_CH4
PB8	PB8	DATA24	COMP_OUT	TMR3_CH4N
PB9	PB9	DATA25	nSIRIN	EXT_INT4
PB10	PB10	DATA26	EXT_INT2	Nsirout1
<b>Порт С</b>				
PC0	PC0	-	SCL1	SSP2_FSS
PC1	PC1	OE	SDA1	SSP2_CLK
PC2	PC2	WE	TMR3_CH1	SSP2_RXD
<b>Порт D</b>				
PD0	PD0	TMR1_CH1N	UART2_RXD	TMR3_CH1
PD1	PD1	TMR1_CH1	UART2_TXD	TMR3_CH1N
PD2	PD2	BUSY1	SSP2_RXD	TMR3_CH2
PD3	PD3	-	SSP2_FSS	TMR3_CH2N
PD4	PD4	TMR1_ETR	nSIRIOUT2	TMR3_BLK
PD5	PD5	CLE	SSP2_CLK	TMR2_ETR
PD6	PD6	ALE	SSP2_TXD	TMR2_BLK
PD7	PD7	TMR1_BLK	nSIRIN2	UART1_RXD
<b>Порт E</b>				
PE0	PE0	ADDR16	TMR2_CH1	CAN1_RX
PE1	PE1	ADDR17	TMR2_CH1N	CAN1_TX
PE2	PE2	ADDR18	TMR2_CH3	TMR3_CH1
PE3	PE3	ADDR19	TMR2_CH3N	TMR3_CH1N
PE6	PE6	ADDR20	TMR2_CH4N	TMR3_CH2
PE7	PE7	ADDR21	TMR2_BLK	TMR3_CH2N
<b>Порт F</b>				
PF0	PF0	ADDR0	SSP1_TXD	UART2_RXD
PF1	PF1	ADDR1	SSP1_CLK	UART2_TXD
PF2	PF2	ADDR2	SSP1_FSS	CAN2_RX
PF3	PF3	ADDR3	SSP1_RXD	CAN2_TX
PF4	PF4	ADDR4	-	-
PF5	PF5	ADDR5	-	-
PF6	PF6	ADDR6	TMR1_CH1	-

**MDR\_PORTx->PD**

Регистр фактически содержит два отдельных 16-разрядных регистра. С его помощью можно включать или отключать триггер Шмитта на входных линиях портов, а также переключать режим работы выходного драйвера. В регистр можно записывать данные и считывать из него данные. Значение регистра после сброса: 0. Би-

ты с 0 по 15 позволяют управлять триггером Шмитта. Бит 0 соответствует линии Px0, бит 1 соответствует линии Px1, и так далее до 15 бита. При значении «0» триггер Шмитта отключен, при «1» – триггер Шмитта включен.

Биты с 16 по 31 позволяют переключать режим работы выхода. Бит 16 соответствует линии Px0, бит 1 соответствует линии Px1, и так далее до

15 бита. При значении «0» выход настроен как управляемый драйвер, при «1» – выход настроен как открытый исток.

У рассматриваемого микроконтроллера достаточно сложная система тактирования. Работа с ней будет рассмотрена подробно в последующих статьях. Чтобы можно было выполнять примеры программ, непосредственно в начале функции main следует вставлять несколько строк для настройки встроенного тактового генератора:

```
// использовать генератор высокой частоты HSE 8 МГц как источник тактовой частоты
MDR_RST_CLK->HS_CONTROL = 0x1; // разрешить работу генератора HSE
while (MDR_RST_CLK->CLOCK_STATUS == 0x00) __NOP(); // ждать запуска HSE
MDR_RST_CLK->CPU_CLOCK = 0x102; // переключить HSE в 8 МГц
```

Важно помнить, что для включения в работу порта необходимо предварительно в регистре разрешения тактирования периферийных блоков разрешить его тактирование. Соответствие битов регистра MDR\_RST\_CLK->PER\_CLOCK и названия порта:

- 21 – PORT A,
- 22 – PORT B,
- 23 – PORT C,
- 24 – PORT D,
- 25 – PORT E.

Например, для включения тактирования порта С следует указать:

```
MDR_RST_CLK->PER_CLOCK |= (1UL << 23); // Разрешить тактирование PORT C
```

Пример настройки линий PC0 и PC1 порта С для работы на выход:

```
// настроить выходы PC0 и PC1 порта С как выходы
MDR_PORTC->OE |= ((1 << 0) | (1 << 1)); // направление - выход
MDR_PORTC->FUNC &= ~( (3 << 0*2) | (3 << 1*2) ); // функция - порт
MDR_PORTC->ANALOG |= ((1 << 0) | (1 << 1)); // цифровой режим
MDR_PORTC->PWR &= ~( (3 << 0*2) | (3 << 1*2) );
MDR_PORTC->PWR |= ((1 << 0*2) | (1 << 1*2)); // медленный фронт
```

После этой настройки порта можно формировать логические уровни напряжения на линиях PC0 и PC1. Для этого следует использовать команды:

```
MDR_PORTC->RXTX &= ~1UL << 0;
// выставить значение «0» на PC0
MDR_PORTC->RXTX &= ~1UL << 1;
// выставить значение «0» на PC1

MDR_PORTC->RXTX |= 1UL << 0;
// выставить значение «1» на PC0
MDR_PORTC->RXTX |= 1UL << 1;
// выставить значение «1» на PC1
```

Для удобства определим маски для выводов с помощью #define. Предположим, к линии PC1 подключена схема, управляющая включением лампы. Если в начале программы записать #define Lamp1 1UL << 1, то далее в программе можно указать:

```
MDR_PORTC->RXTX |= Lamp1; //
включить лампу
MDR_PORTC->RXTX &= ~Lamp1 //
выключить лампу
```

В результате программа хорошо читается, и облегчается программирова-

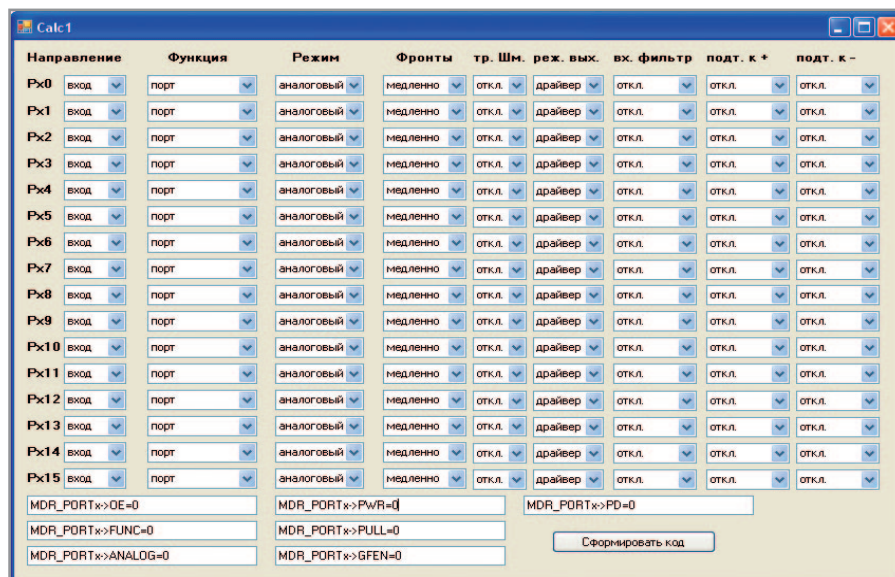


Рис. 2. Окно программы для настройки портов

ние переключения на другую линию порта (достаточно изменить только строчку с #define, остальной текст программы остаётся неизменным). Разумеется, другую линию порта при этом следует настроить для работы на выход.

Для удобства работы автором статьи написана небольшая программа, облегчающая настройку портов. На ри-

сунке 2 показано её окно. Для использования программы необходимо установить требуемые режимы для каждой линии порта, после чего нажать на кнопку «Сформировать код». В полях отобразятся строки кода, которые можно скопировать в свою программу. Программу настройки портов можно скачать с интернет-страницы журнала ([www.soel.ru](http://www.soel.ru)). ©