

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего профессионального образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

ПОРТЫ МИКРОКОНТРОЛЛЕРА 1986ВЕ93У

Методические указания
к выполнению лабораторной работы

Санкт-Петербург
2017

Составители кандидат технических наук доцент *Р.Н. Малаханов*,
Е.В. Савенкова
Рецензент кандидат технических наук доцент *В.К. Пономарёв*

Методические указания знакомят с устройством портов микроконтроллера 1986ВЕ93У производства АО «ПКК Миландр» и принципами их программного управления.

Содержат описание лабораторной работы по дисциплинам «Цифровые вычислительные устройства и микропроцессоры», «Бортовые цифровые вычислительные машины» и «Бортовые вычислительные комплексы».

Предназначены для студентов всех форм обучения по направлению «Приборостроение».

Подготовлены кафедрой аэрокосмических измерительно-вычислительных комплексов и рекомендованы к изданию редакционно-издательским советом Санкт-Петербургского государственного университета аэрокосмического приборостроения.

Список сокращений

АП – адресное пространство

АР – аналоговый режим

АС – аппаратный сброс

АЦП – аналого-цифровой преобразователь

БА – базовый адрес

ВУ – внешнее устройство

ДОП – демонстрационно-отладочная плата

ЕИ – единичный индикатор

ИМС – интегральная микросхема

МК – микроконтроллер

МП – микропроцессор

ОЗУ – оперативное запоминающее устройство

ПВВ – порт ввода-вывода

РУП – регистр управления портом

РЦВВ – режим циклического включения-выключения

СИД – светоизлучающий диод

ЦАП – цифро-аналоговый преобразователь

ЦВВ – цифровой ввод-вывод

ЦР – цифровой режим

Лабораторная работа

1. Методические указания

Цель работы заключается в ознакомлении с функциональным устройством портов МК 1986BE93У и принципами их программного управления.

Основные термины и определения

Портом называется устройство, которое позволяет МП обмениваться информацией с ВУ. В зависимости от направления передачи информации по отношению к МП различают порты ввода, вывода и ввода-вывода информации. ПВВ информации ещё называют двунаправленными портами.

В зависимости от типов сигналов, с помощью которых производится обмен информацией, порты подразделяются на цифровые, аналоговые и смешанные. Термин «смешанный порт» является переводом с англ. «*mixed signal port*». Если обмен информацией производится в каком-нибудь цифровом коде, то такие порты называются цифровыми портами. В современных микропроцессорных системах обмен информацией производится, как правило, в двоичном коде. Если обмен информацией производится с помощью аналоговых сигналов, то такие порты называются аналоговыми. Смешанные порты могут работать как цифровые и как аналоговые. Выбор режима работы порта задаётся отдельным сигналом управления.

По способу передачи числа в двоичном коде порты подразделяются на параллельные и последовательные. В параллельных портах ввод или вывод всех разрядов числа в двоичном коде производится одновременно. Для этого каждому разряду двоичного числа соответствует проводник, по которому электрический сигнал, уровень напряжения которого соответствует логическому значению разряда, поступает на ВУ. В последовательных портах все

разряды двоичного числа передаются один за другим последовательно во времени.

В микроконтроллере ПВВ являются встроенными и могут управляться программно. Группу выводов МК объединяют в один порт и присваивают ему уникальное имя. За этим портом также закрепляются несколько РУП. Как правило, количество выводов, объединяемых в порт, кратно восьми, т.е. одному байту. В 8-разрядных МК в порт объединяется восемь выводов, а в 16- и 32-разрядных МК – 16 выводов. Но если у корпуса МК мало выводов, то количество выводов в порте может быть меньше восьми.

В лабораторной работе будет производится управление ЕИ, поэтому рассмотрим основные термины и определения в соответствии с ГОСТ 25066-91 [1].

Знакосинтезирующий индикатор - прибор, в котором информация, предназначенная для зрительного восприятия, отображается с помощью одного или совокупности дискретных элементов. *Единичный (знакосинтезирующий) индикатор* – знакосинтезирующий индикатор, состоящий из одного элемента изображения и предназначенный для отображения информации в виде точки или геометрической фигуры. В современной электронике ЕИ выполняются, как правило, на основе СИД.

Порты микроконтроллера 1986ВЕ93У

МК имеет шесть встроенных портов и они все являются смешанными ПВВ [2]. Порты обозначаются латинскими буквами *A, B, C, D, E* и *F*. Порт *A* имеет 8 разрядов; порт *B* – 7 разрядов; порт *C* – 1 разряд; порты *E* и *D* - по 4 разряда; порт *F* – 6 разрядов.

Аппаратно порты выполнены как параллельные, поэтому каждый разряд порта имеет свой вывод в корпусе МК. Для обозначения вывода порта к названию порта добавляются слева буква *P*, а справа - порядковый номер раз-

ряда, которому соответствует этот вывод, например, второй разряд порта *A* обозначается как *PA2*.

Микроконтроллер 1986BE93У выпускается в корпусе H16.48-1В [2], расположение выводов которого приведено на рис. 1. К выводам *Ucc* и *GND* подключается источник электрического питания 3,3 В. Описание выводов, которые не принадлежат портам, приведено в [2].

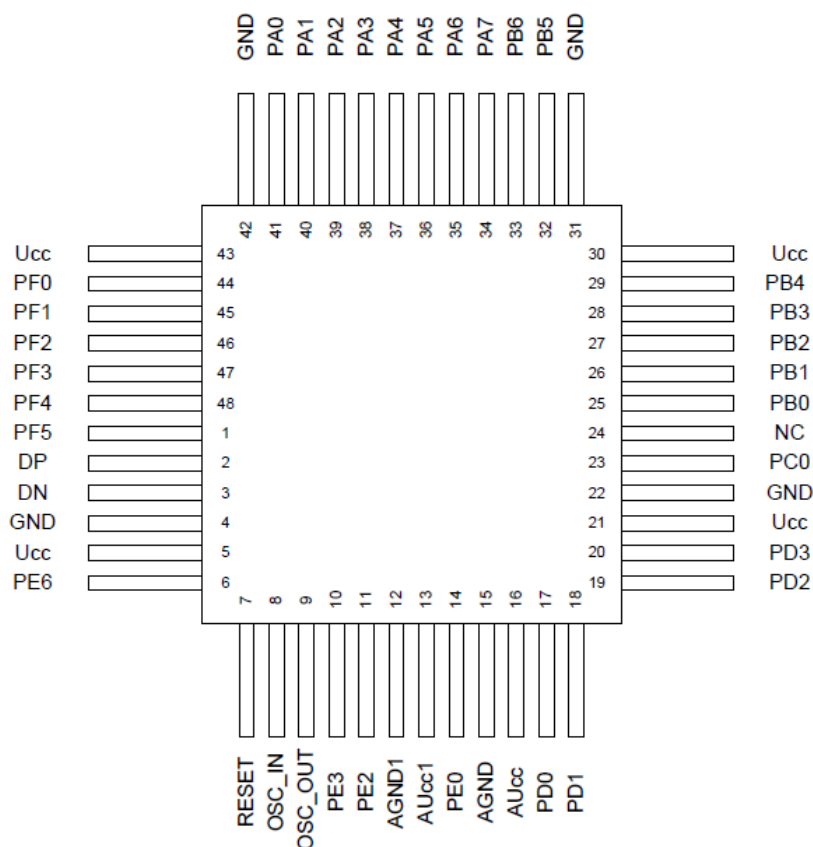


Рисунок 1 – Расположение выводов МК 1986BE93У.

Функциональная схема одного разряда порта приведена на рис. 2. Вывод МК обозначен на рисунке справа как прямоугольник с надписью «*PAD*». Все элементы схемы, которые находятся слева от этого вывода, встроены в МК. Два диода предназначены для ограничения входного напряжения. Резистор *ESD R* предназначен для ограничения входного тока. Обозначение «*ESD*» является сокращением от англ. «*Electric Static Discharge*» - электрический статический разряд. Два проводника, обозначенные как «*To Analog*», являются входом и выходом для аналоговых модулей. Два необозначенных резистора

являются подтягивающими: верхний подтягивает к напряжению электрического питания, а нижний – к потенциалу «земли». Сигналы *PULL UP* и *PULL DOWN* предназначены для управления этими резисторами. Буферы *TX* и *RX* являются передающим и приёмным соответственно и предназначены для приёма и передачи цифровых сигналов. Сигнал *Analog* управляет работой буферов. Сигнал *GFEN* управляет включением фильтра, который фильтрует импульсы, длительность которых меньше 10 нс. Два мультиплексора *MUX* предназначены для передачи цифровых сигналов. Сигнал *PD* управляет выходным драйвером.

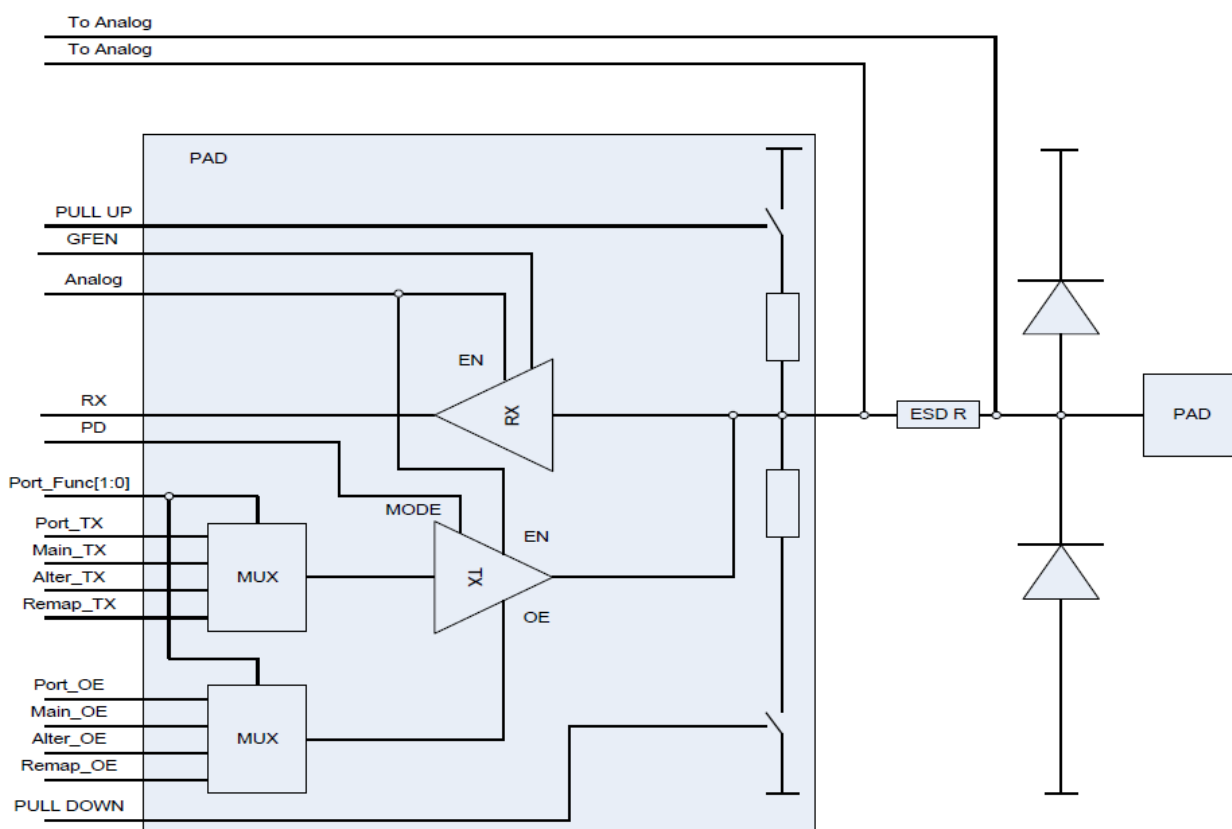


Рисунок 2 – Функциональная схема одного вывода порта.

АП портов распределено на ОЗУ микроконтроллера. БА портов приведены в табл. 1. БА является первым адресом АП порта и от его значения отсчитывается адрес РУП. В [2] для указания порта используется обозначение *MDR_PORTx*, где вместо буквы *x* подставляется буква от *A* до *F*, обозначающая порт.

Таблица 1 – Базовые адреса портов

Базовый адрес	Название порта	Описание
400A8000h	<i>MDR_PORTA</i>	Порт <i>A</i>
400B0000h	<i>MDR_PORTB</i>	Порт <i>B</i>
400B8000h	<i>MDR_PORTC</i>	Порт <i>C</i>
400C0000h	<i>MDR_PORTD</i>	Порт <i>D</i>
400C8000h	<i>MDR_PORTE</i>	Порт <i>E</i>
400E8000h	<i>MDR_PORTF</i>	Порт <i>F</i>

Функциональное назначение РУП и смещения их адресов относительно БА приведены в табл. 2. Все РУП являются 32-разрядными, поэтому смещения их адресов идут с шагом, равным четырём байтам. Для того, чтобы получить адрес РУП, надо к БА порта, приведённому в табл. 1, прибавить смещение РУП из табл. 2. Например, для получения адреса регистра *ANALOG* порта *F* надо к БА порта *F* 400E8000h прибавить смещение 0Ch регистра *ANALOG* и получится адрес 400E800Ch. Регистр *RXTX* имеет смещение 0, поэтому его адрес совпадает с БА порта. В квадратных скобках после названия регистров указывается диапазон используемых разрядов этого регистра. В [2] для указания РУП используется обозначение *MDR_PORTx->имя регистра*, где вместо буквы *x* подставляется буква, обозначающая порт, например, *MDR_PORTF->ANALOG* - регистр *ANALOG* порта *F*.

Для программного управления и настройки режимов работы отдельного вывода МК за ним закрепляется определённый разряд во всех регистрах управления того порта, к которому относится вывод. Например, для управления и настройки вывода *PA0* предназначены все нулевые разряды всех регистров управления порта *A*; для управления и настройки вывода *PB4* предназначены все четвёртые разряды всех регистров управления порта *B* и т.д.

Таблица 2 – Регистры управления портом

Смещение	Имя регистра	Функциональное назначение
00h	<i>RXTX</i> [15:0]	Данные, которые записываются или считываются из порта
04h	<i>OE</i> [15:0]	Настройка направления передачи данных: ввод данных или вывод
08h	<i>FUNC</i> [31:0]	Настройка порта в ЦР
0Ch	<i>ANALOG</i> [15:0]	Выбор режима работы: аналоговый или цифровой
10h	<i>PULL</i> [31:0]	Управление включением подтягивающих резисторов
14h	<i>PD</i> [31:0]	Управление выходным драйвером
18h	<i>PWR</i> [31:0]	Управление длительностью фронтов генерируемых сигналов
1Ch	<i>GFEN</i> [15:0]	Управление входным фильтром

Одни и те же выводы МК могут использоваться аналоговыми (АЦП, ЦАП, компаратор) и цифровыми встроенными модулями, поэтому для настройки работы вывода с необходимыми встроенными модулями предназначены регистры *ANALOG* и *FUNC*. Регистр *ANALOG* предназначен для выбора режима работы выводов МК между аналоговым или цифровым режимами, а регистр *FUNC* - для последующей настройки работы выводов в ЦР. Если вывод работает в AP, то значение в регистре *FUNC* на его работу не влияет.

Формат регистра *ANALOG* приведён в табл. 3, а описание функционального назначения его разрядов – в табл. 4. В строке «Номер» приведены диапазоны порядковых номеров разрядов регистра. В строке «Доступ» указана возможность проведения операций записи или чтения разрядов регистра. Сокращение «R/W» (от англ. *Read/Write*) указывает на то, что биты с 15 по 0

доступны для записи и чтения. Сокращение «*U*» указывает на то, что разряды с 31 по 16 недоступны для записи. В строке «Сброс» указано значение, которое присваивается всем разрядам диапазона после АС. Если какой-нибудь разряд *ANALOG EN* сброшен, то соответствующий ему вывод МК работает в АР, иначе – в ЦР. Например, если в:

- регистр *ANALOG* порта *D* записано число $0000000Bh$, где младший байт равен $0Bh=00001011_2$;

- регистр *ANALOG* порта *E* записано число $00000044h$, где младший байт равен $44h=01000100_2$,

то выходы *PD0*, *PD1*, *PD3*, *PE2* и *PE6* будут работать в ЦР, а выходы *PD2*, *PE0* и *PE3* – в АР. У этого МК выходы *PE1*, *PE4* и *PE5* не реализованы, поэтому в первый, четвёртый и пятый разряды всех РУП *E* можно записывать любые значения. После АС все выходы МК настраиваются на работу в АР.

Таблица 3 – Формат регистра *ANALOG*

Номер	31...16	15...0
Доступ	<i>U</i>	<i>R/W</i>
Сброс	0	0
Имя разряда	<i>ANALOG EN</i> [15:0]	

Таблица 4 – Описание разрядов регистра *ANALOG*

№ разряда	Имя разряда	Описание
31...16		Зарезервировано
15...0	<i>ANALOG EN</i> [15:0]	Режим работы вывода порта: 0 – аналоговый; 1 – цифровой

Регистр *FUNC* предназначен для настройки выводов в ЦР для выполнения трёх функций: основной, альтернативной или переопределённой. Если ни одна из функций не выбрана, то вывод будет работать в режиме ЦВВ. Формат регистра *FUNC* приведён в табл. 5, а описание его разрядов – в табл. 6. Для настройки одного вывода порта предназначены два разряда настройки *MODE*. Например, для настройки вывода *PA0* предназначены разряды 1 и 0 (*MODE0*) регистра *FUNC* порта *A*; для настройки вывода *PA1* предназначены разряды 3 и 2 (*MODE1*) этого же регистра и т.д. Каждый вывод порта может выполнять одну из трёх функций или работать в режиме ЦВВ.

Таблица 5 – Формат регистра *FUNC*

Номер	31	30	...	3	2	1	0
Доступ	<i>R/W</i>	<i>R/W</i>	...	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
Сброс	0	0	...	0	0	0	0
Имя разряда	<i>MODE15</i> [1:0]		...	<i>MODE1</i> [1:0]		<i>MODE0</i> [1:0]	

Таблица 6 – Описание разрядов регистра *FUNC*

№ разряда	Имя разряда	Описание
31...2	<i>MODE_x</i>	Аналогично <i>MODE0</i> для остальных разрядов
1...0	<i>MODE0</i> [1:0]	Режим работы вывода порта: 00 ₂ – цифровой ввод-вывод; 01 ₂ – основная функция; 10 ₂ – альтернативная функция; 11 ₂ – переопределённая функция

Возможности выводов МК по выполнению различных функций приведены в таблицах с 7 по 12. Например, вывод *PE0* может использоваться (см. табл. 11):

- модулем ЦАП (*DAC2_OUT*, *Digital-to-Analog Converter Output*);
- модулем таймера 2 для генерации выходного сигнала первого канала (*TMR2_CH1*, *Timer 2 Channel 1*);
- модулем *CAN1* для приёма данных по интерфейсу *CAN* (*CAN1_RX*);
- как нулевой разряд порта *E* в режиме ЦВВ.

Все обозначения, приведённые в таблицах с 7 по 12, более подробно описаны в [2].

После АС все выводы МК настраиваются на работу в режиме ЦВВ.

Таблица 7 – Описание функций выводов порта *A*

Вы- вод	Аналоговая функция	Цифровая функция (<i>ANALOG EN = 1</i>)			
		Цифровой ввод-вывод	Основная	Альтерна- тивная	Переопреде- лённая
<i>PA0</i>		<i>PA0</i>		<i>EXT_INT1</i>	
<i>PA1</i>		<i>PA1</i>		<i>TMR1_CH1</i>	<i>TMR2_CH1</i>
<i>PA2</i>		<i>PA2</i>		<i>TMR1_CH1N</i>	<i>TMR2_CH1N</i>
<i>PA3</i>		<i>PA3</i>		<i>TMR1_CH2</i>	<i>TMR2_CH2</i>
<i>PA4</i>		<i>PA4</i>		<i>TMR1_CH2N</i>	<i>TMR2_CH2N</i>
<i>PA5</i>		<i>PA5</i>		<i>TMR1_CH3</i>	<i>TMR2_CH3</i>
<i>PA6</i>		<i>PA6</i>		<i>CAN1_TX</i>	<i>UART1_RXD</i>
<i>PA7</i>		<i>PA7</i>		<i>CAN1_RX</i>	<i>UART1_TXD</i>

Таблица 8 – Описание функций выводов порта *B*

Вы-вод	Аналоговая функция	Цифровая функция (<i>ANALOG EN = 1</i>)			
		Цифровой ввод-вывод	Основная	Альтернативная	Переопределённая
<i>PB0</i>		<i>PB0, JA_TDO</i>		<i>TMR3_CH1</i>	<i>UART1_RXD</i>
<i>PB1</i>		<i>PB1, JA_TMS</i>		<i>TMR3_CH1N</i>	<i>UART1_TXD</i>
<i>PB2</i>		<i>PB2, JA_TCK</i>		<i>TMR3_CH2</i>	<i>CAN1_TX</i>
<i>PB3</i>		<i>PB3, JA_TDI</i>		<i>TMR3_CH2N</i>	<i>CAN1_RX</i>
<i>PB4</i>		<i>PB4, JA_TRST</i>		<i>TMR3_BLK</i>	<i>TMR3_ETR</i>
<i>PB5</i>		<i>PB5</i>		<i>UART1_RXD</i>	<i>TMR3_CH3</i>
<i>PB6</i>		<i>PB6</i>		<i>UART1_TXD</i>	<i>TMR3_CH3N</i>

Таблица 9 – Описание функций выводов порта *C*

Вы-вод	Аналоговая функция	Цифровая функция (<i>ANALOG EN = 1</i>)			
		Цифровой ввод-вывод	Основная	Альтернативная	Переопределённая
<i>PC0</i>		<i>PC0</i>		<i>SCL1</i>	<i>SSP2_FSS</i>

Формат регистра *RXTX* приведён в табл. 13, а описание его разрядов - в табл. 14. Если порт работает в режиме цифрового вывода, то запись в определённый разряд регистра *RXTX* значения 0 или 1 приведёт к установке на выводе МК, которому соответствует этот разряд, логического уровня напряжения, соответствующего записанному значению. Если в разряд записано значение 1, то на выводе МК установится напряжение, значение которого близко к значению напряжения электрического питания МК 3,3 В. Если в разряд записано значение 0, то на выводе МК установится напряжение, близкое к нулю. Например, если порт *A* работает в режиме цифрового вывода, то запись в регистр *RXTX* значения

$$00006FAC_h = 0000\ 0000\ 0000\ 0000\ 0110\ 1111\ 1010\ 1100_2$$

приведёт к установке на выводах МК *PA0*, *PA1*, *PA4*, *PA6* напряжения, значение которого близко к нулю, а на выводах *PA2*, *PA3*, *PA5*, *PA7* напряжения, значение которого близко к 3,3 В. Запись любого значения в биты с 15 по 8 (в примере *6Fh*) не приведёт к изменению выходных сигналов, потому что у этого МК разряды с 15 по 8 не выведены на выводы корпуса.

Таблица 10 – Описание функций выводов порта *D*

Вы- вод	Аналоговая функция	Цифровая функция (<i>ANALOG EN = 1</i>)			
		Цифр. ввод- вывод	Основная	Альтерна- тивная	Переопреде- лённая
<i>PD0</i>	<i>ADC0_REF+</i>	<i>PD0</i>	<i>TMR1_CH1N</i>	<i>UART2_RXD</i>	<i>TMR3_CH1</i>
<i>PD1</i>	<i>ADC1_REF-</i>	<i>PD1</i>	<i>TMR1_CH1</i>	<i>UART2_TXD</i>	<i>TMR3_CH1N</i>
<i>PD2</i>	<i>ADC2</i>	<i>PD2</i>		<i>SSP2_RXD</i>	<i>TMR3_CH2</i>
<i>PD3</i>	<i>ADC3</i>	<i>PD3</i>		<i>SSP2_FSS</i>	<i>TMR3_CH2N</i>

Таблица 11 – Описание функций выводов порта *E*

Вы- вод	Аналоговая функция	Цифровая функция (<i>ANALOG EN = 1</i>)			
		Цифр. ввод- вывод	Основ- ная	Альтерна- тивная	Переопреде- лённая
<i>PE0</i>	<i>DAC2_OUT</i>	<i>PE0</i>		<i>TMR2_CH1</i>	<i>CAN1_RX</i>
<i>PE2</i>	<i>COMP_IN1</i>	<i>PE2</i>		<i>TMR2_CH3</i>	<i>TMR3_CH1</i>
<i>PE3</i>	<i>COMP_IN2</i>	<i>PE3</i>		<i>TMR2_CH3N</i>	<i>TMR3_CH1N</i>
<i>PE6</i>	<i>OSC_IN32</i>	<i>PE6</i>		<i>CAN2_RX</i>	<i>TMR3_CH3</i>

Таблица 12 – Описание функций выводов порта *F*

Вы- вод	Аналого- вая функ- ция	Цифровая функция (<i>ANALOG EN = 1</i>)			
		Цифровой ввод-вывод	Основная	Альтерна- тивная	Переопреде- лённая
<i>PF0</i>		<i>PF0</i>		<i>SSP1TXD</i>	<i>UART2_RXD</i>
<i>PF1</i>		<i>PF1</i>		<i>SSP1CLK</i>	<i>UART2_TXD</i>
<i>PF2</i>		<i>PF2</i>		<i>SSP1FSS</i>	<i>CAN2_RX</i>
<i>PF3</i>		<i>PF3</i>		<i>SSP1RXD</i>	<i>CAN2_TX</i>
<i>PF4</i>		<i>PF4</i>			
<i>PF5</i>		<i>PF5</i>			

Таблица 13 – Формат регистра *RXTX*

Номер	31...16	15...0
Доступ	<i>U</i>	<i>R/W</i>
Сброс	0	0
Имя разряда		<i>PORT RXTX [15:0]</i>

Таблица 14 – Описание разрядов регистра *RXTX*

№ разряда	Имя разряда	Описание
31...16		Зарезервировано
15...0	<i>PORT RXTX [15:0]</i>	Данные, которые записываются или считываются из порта

Если порт работает в режиме цифрового ввода, то значения разрядов регистра *RXTX* отображают логические уровни напряжений, подаваемых ВУ на

выводы МК. Если значение разряда равно 0, то на вывод МК, соответствующий этому разряду, подано значение напряжения, соответствующее логическому нулю. Если значение разряда равно 1, то на вывод МК подано значение напряжения, соответствующее логической единице.

В режиме ЦВВ каждый вывод порта может работать как цифровой ввод или как цифровой вывод. Выбор режима работы определяется программно посредством записи значения в регистр *OE* (сокращение от англ. *Output Enable*), формат которого приведён в табл. 15. Описание разрядов регистра *OE* приведено в табл. 16. Если разряд регистра *OE* сброшен, то соответствующий ему вывод МК будет работать в режиме цифрового ввода; иначе - в режиме цифрового вывода. Например, если в регистр *OE* порта *B* записано число 0000006Ah, где младший байт 6Ah=01101010₂, то выводы МК *PB0*, *PB2*, *PB4* будут работать в режиме цифрового ввода, а выводы *PB1*, *PB3*, *PB5* и *PB6* – в режиме цифрового вывода.

Таблица 15 – Формат регистра *OE*

Номер	31...16	15...0
Доступ	U	R/W
Сброс	0	0
Имя разряда		<i>PORT OE</i> [15:0]

Регистр *PULL* предназначен для управления подключением подтягивающих резисторов. Формат регистра приведён в табл. 17, а описание его разрядов – в табл. 18. Разряды *PULL UP* предназначены для включения резисторов, подтягивающих к значению напряжения электрического питания, а разряды *PULL DOWN* - для включения резисторов, подтягивающих к потенциалу «земли». Для подключения подтягивающего резистора следует установить соответствующий разряд. Например, для подключения резисторов:

- подтягивающих к значению напряжения электрического питания на выводах *PA2* и *PA3*;

- подтягивающих к потенциалу «земли» на вывод *PA6*,
следует в регистр *PULL* порта *A* записать значение:

$$000C0040h = 0000\ 0000\ 0000\ 1100\ 0000\ 0000\ 0100\ 0000_2.$$

После АС все резисторы отключены. Номинальные значения резисторов составляют около 50 кОм [2].

Таблица 16 – Описание разрядов регистра *OE*

№ разряда	Имя разряда	Описание
31...16		Зарезервировано
15...0	<i>PORT OE</i> [15:0]	Настройка работы порта в режиме ЦВВ: 1 – цифровой вывод; 0 – цифровой ввод

Таблица 17 – Формат регистра *PULL*

Номер	31...16	15...0
Доступ	<i>R/W</i>	<i>R/W</i>
Сброс	0	0
Имя разряда	<i>PULL UP</i> [15:0]	<i>PULL DOWN</i> [15:0]

Регистр *PD* (*Port Driver*) предназначен для настройки режима работы выходного драйвера порта. Формат регистра приведён в табл. 19, а описание его разрядов – в табл. 20.

Таблица 18 – Описание разрядов регистра *PULL*

№ разряда	Имя разряда	Описание
31...16	<i>PULL UP</i> [15:0]	Включение резистора, подтягивающего к значению напряжения электрического питания: 0 – резистор не подключён; 1 – резистор подключён
15...0	<i>PULL DOWN</i> [15:0]	Включение резистора, подтягивающего к потенциалу «земли»: 0 – резистор не подключён; 1 – резистор подключён

Регистр *PWR* (сокращение от англ. *Power*) позволяет установить длительности фронтов и срезов импульсов, генерируемых портом в ЦР. Формат регистра приведён в табл. 21, а описание его разрядов – в табл. 22. Длительность фронтов и срезов импульсов влияет на ширину спектра и интенсивность электромагнитных излучений, поэтому, если не требуется формировать импульсы с короткими фронтами и срезами, то следует устанавливать медленный фронт для вывода МК с целью уменьшения ширины спектра электромагнитного излучения. Для настройки длительности фронтов сигналов для одного вывода МК предназначены два разряда регистра. Например, для настройки выводов МК:

- *PA0* и *PA1* на длительность фронта импульса 100 нс;
- *PA2* и *PA3* на длительность фронта импульса 20 нс;
- *PA4* и *PA5* на длительность фронта импульса 10 нс,

в регистр *PWR* порта *A* следует записать значение:

$0000FA5h = 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 1010\ 0101_2$;

передатчики выводов *PA6* и *PA7* будут отключены.

Таблица 19 – Формат регистра *PD*

Номер	31...16	15...0
Доступ	<i>R/W</i>	<i>R/W</i>
Сброс	0	0
Имя разряда	<i>PORT SHM</i> [15:0]	<i>PORT PD</i> [15:0]

Таблица 20 – Описание разрядов регистра *PD*

№ разряда	Имя разряда	Описание
31...16	<i>PORT SHM</i> [15:0]	Включение триггера Шмидта на входе: 0 – триггер выключён, гистерезис 200 мВ; 1 – триггер включён, гистерезис 400 мВ
15...0	<i>PORT PD</i> [15:0]	Режим работы выхода: 0 – управляемый драйвер; 1 – открытый сток

Таблица 21 – Формат регистра *PWR*

Номер	31	30	...	3	2	1	0
Доступ	<i>R/W</i>	<i>R/W</i>	...	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
Сброс	0	0	...	0	0	0	0
Имя разряда	<i>PWR15</i> [1:0]		...	<i>PWR1</i> [1:0]		<i>PWR0</i> [1:0]	

Регистр *GFEN* предназначен для включения входного фильтра для фильтрации импульсов, длительность которых меньше 10 нс. Это позволяет обеспечить помехоустойчивый приём цифровых сигналов при наличии им-

пульсных помех. Формат регистра приведён в табл. 23, а описание разрядов регистра – в табл. 24. После АС фильтры всех выводов портов выключены.

Таблица 22 – Описание разрядов регистра *PWR*

№ разряда	Имя разряда	Описание
31...2	<i>PWR_x</i>	Аналогично <i>PWR0</i> для остальных разрядов
1...0	<i>PWR0</i> [1:0]	Длительность фронта импульса: 00 ₂ – зарезервировано (передатчик отключён); 01 ₂ – медленный фронт 100 нс; 10 ₂ – быстрый фронт 20 нс; 11 ₂ – максимально быстрый фронт 10 нс

Таблица 23 – Формат регистра *GFEN*

Номер	31...16	15...0
Доступ	<i>U</i>	<i>R/W</i>
Сброс	0	0
Имя разряда		<i>GFEN</i> [15:0]

Таблица 24 – Описание разрядов регистра *GFEN*

№ разряда	Имя разряда	Описание
31...16		Зарезервировано
15...0	<i>GFEN</i> [15:0]	Управление включением входного фильтра: 0 – входной фильтр выключен; 1 – входной фильтр включён

Схемы управления светоизлучающими диодами

Схемы подключения СИД непосредственно к выводам ИМС приведены на рис. 3. На схеме рис. 3, а включение СИД происходит при логическом единичном уровне напряжения на выходе ИМС. Значение сопротивления токоограничивающего резистора определяется из выражения

$$R = (U_{\text{вых}}^1 - U_{\text{пр}}) / I_{\text{пр}},$$

где $U_{\text{вых}}^1$ — напряжение логической единицы на выходе ИМС; $U_{\text{пр}}$ - постоянное прямое напряжение на СИД; $I_{\text{пр}}$ - постоянный прямой ток СИД.

На рис. 3, б включение СИД происходит при нулевом уровне напряжения на выходе ИМС. Значение сопротивления токоограничивающего резистора определяется выражением

$$R = (U_{\text{пит}} - U_{\text{пр}} - U_{\text{вых}}^0) / I_{\text{пр}},$$

где $U_{\text{пит}}$ — напряжение источника электрического питания; $U_{\text{вых}}^0$ — нулевой логический уровень выходного напряжения ИМС.

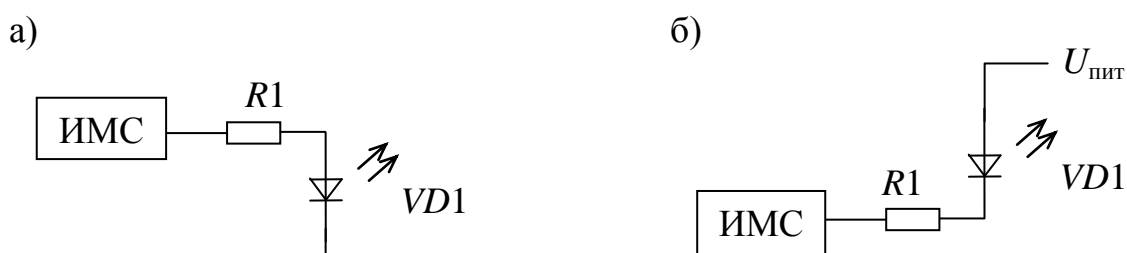


Рисунок 3 – Схемы подключения СИД к выводам ИМС.

Подключение кнопок

Варианты схем подключения кнопок к ИМС приведены на рис. 4. Определение состояния кнопки может быть проведено по активным низкому (рис. 4, а) и высокому (рис. 4, б) логическим уровням. Когда кнопка не нажата, то на вход ИМС поступает высокий (рис. 4, а) или низкий (рис. 4, б) логические уровни. При нажатии кнопки на вход ИМС поступает низкий (рис. 4, а) или высокий (рис. 4, б) логические уровни. Резистор $R1$ необходим для предотвращения короткого замыкания при нажатии кнопки.

Контакты кнопок состоят из упругих элементов, поэтому при нажатии кнопки замыкание контактов происходит не мгновенно, а в течении определённого времени из-за механических упругих колебаний (соударений) контактов. В течении этого времени происходят многочисленные замыкания и размыкания контактов, поэтому сигнал от кнопки может быть прочитан как случайная последовательность нулей и единиц. Такие же процессы происходят и при отпускании кнопки. Это явление называется дребезгом контактов. Из-за множественного случайного изменения уровня сигнала на входе ИМС может быть определено, что пользователь нажал кнопку несколько раз, хотя произошло только её однократное нажатие. Для устранения дребезга контактов в схему включается интегрирующий конденсатор $C1$ (рис. 4, в), который сглаживает случайную импульсную последовательность.

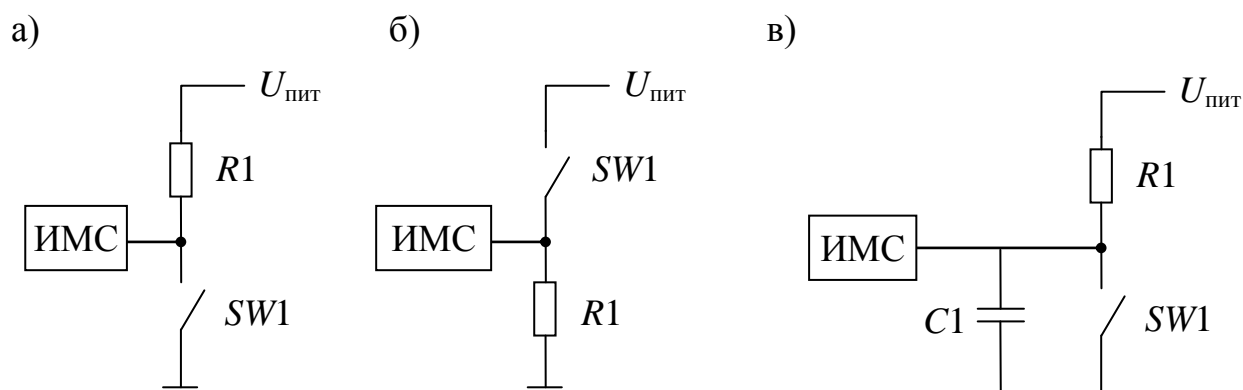


Рисунок 4 - Функциональные схемы подключения кнопок.

Программное обеспечение

Производитель МК предоставляет библиотеку [3], которая содержит файлы с исходными текстами функций на языке программирования Си для различных встроенных периферийных модулей.

Во включаемом файле *stdint.h* определяются три новых типа *uint8_t*, *uint16_t* и *uint32_t*:

```
typedef unsigned char uint8_t;
typedef unsigned short int uint16_t;
typedef unsigned int uint32_t;
```

которые в дальнейшем используются при программировании портов.

Во включаемом файле *MDR32Fx.h* приведены определения констант, задающих БА портов. Константы имеют имена формата *MDR_PORTX_BASE*, где вместо буквы *X* следует подставить букву имени порта. Например, константа, задающая БА порта *A*, определяется строкой

```
#define MDR_PORTA_BASE (0x400A8000)
```

Таким же образом задаются остальные константы БА всех портов, которые приведены в табл. 1.

В этом же файле объявляется новый тип *MDR_PORT_TypeDef*, который представляет собой структуру, поля которой совпадают с названиями РУП:

```
typedef struct
{
    __IO uint32_t RXTX;
    __IO uint32_t OE;
```

```

__IO uint32_t FUNC;
__IO uint32_t ANALOG;
__IO uint32_t PULL;
__IO uint32_t PD;
__IO uint32_t PWR;
__IO uint32_t GFEN;
} MDR_PORT_TypeDef;.

```

Константа `__IO` определена в файле `core_cm3.h`:

```
#define __IO volatile.
```

Также объявляются указатели на тип `MDR_PORT_TypeDef` формата `MDR_PORTX`, где вместо буквы `X` следует подставить букву названия порта. Указателям присваиваются значения констант `MDR_PORTX_BASE`, в результате чего указатели `MDR_PORTX` указывают на БА портов. Например, указатель `MDR_PORTA` определяется как

```
#define MDR_PORTA ((MDR_PORT_TypeDef*) MDR_PORTA_BASE).
```

Указатели `MDR_PORTX` можно использовать для обращений к РУП. Например, запись `MDR_PORTA->ANALOG=4`; позволяет записать число 4 в регистр `ANALOG` порта `A`. В этой записи обращение производится к полю `ANALOG` структуры `MDR_PORT_TypeDef` с помощью указателя `MDR_PORTA` и так как адрес начала структуры (`MDR_PORTA=400A8000h`) совпадает с началом АП порта и поля структуры имеют такой же размер как и размер РУП, то адрес поля `ANALOG` структуры будет совпадать с адресом одноимённого регистра порта `A`.

Обращаться к РУП можно также с помощью указателей, вычисляемых с помощью арифметических операций, осуществляемых над ними. Например, следующий программный код:

```
unsigned int *p = (unsigned int*) MDR_PORTA_BASE; // p = 400A8000h.
p += 3; // p = 400A800Ch – адрес регистра ANALOG
*p = 4;
```

также производит запись числа 4 в регистр *ANALOG* порта *A*. Из сравнения последнего кода и записи *MDR_PORTA->ANALOG=4;* видно, что последняя является более наглядной.

Все декларации и определения, необходимые для работы с портами, находятся в файле *MDR32F9Qx_port.h*, а реализация функций – в файле *MDR32F9Qx_port.c*.

Во всех функциях, предназначенных для работы с портами и описываемых ниже, первым входным параметром является указатель типа *MDR_PORT_TypeDef** формата *MDR_PORTX* для указания порта, с регистрами которого следует производить действия.

Функция *void PORT_DeInit(MDR_PORT_TypeDef*)* записывает во все РУП нули. Например, в результате вызова функции

```
PORT_DeInit(MDR_PORTA);
```

во все регистры управления порта *A* будут записаны нули.

Функция *void PORT_Init (MDR_PORT_TypeDef* PORTx, const PORT_InitTypeDef* PORT_InitStruct)* предназначена для настройки порта. Тип *PORT_InitTypeDef* определён как

```
typedef struct
```

```
{
```

```

uint16_t PORT_Pin;

PORT_OE_TypeDef PORT_OE;

PORT_PULL_UP_TypeDef PORT_PULL_UP;

PORT_PULL_DOWN_TypeDef PORT_PULL_DOWN;

PORT_PD_SHM_TypeDef PORT_PD_SHM;

PORT_PD_TypeDef PORT_PD;

PORT_GFEN_TypeDef PORT_GFEN;

PORT_FUNC_TypeDef PORT_FUNC;

PORT_SPEED_TypeDef PORT_SPEED;

PORT_MODE_TypeDef PORT_MODE;

} PORT_InitTypeDef;.

```

Все поля предназначены для настройки РУП.

Поле *PORT_Pin* предназначено для указания выводов порта, которые надо настроить. Значение *PORT_Pin* задаётся в виде маски: разряды, соответствующие выводам МК, которые надо настроить, устанавливаются. Для удобства составления маски определены константы:

```

#define PORT_Pin_0 0x0001U
#define PORT_Pin_1 0x0002U
#define PORT_Pin_2 0x0004U
#define PORT_Pin_3 0x0008U
#define PORT_Pin_4 0x0010U
#define PORT_Pin_5 0x0020U
#define PORT_Pin_6 0x0040U
#define PORT_Pin_7 0x0080U

```

Для разрядов с 8 по 15 точно так же определены константы. Маска составляется посредством применения операции поразрядного ИЛИ над константами. Например, если надо настроить 0, 3 и 4 выходы порта, то маска составляется следующим образом: *PORT_Pin_0|PORT_Pin_3|PORT_Pin_4*.

Тип *PORT_OE_TypeDef* определяется следующим образом:

```
typedef enum
{
    PORT_OE_IN = 0x0,
    PORT_OE_OUT = 0x1
} PORT_OE_TypeDef;
```

Константы *PORT_OE_IN* и *PORT_OE_OUT* предназначены для записи в поле *PORT_OE*, которое в свою очередь предназначено для настройки регистра *OE* порта (см. табл. 2). Тип, имя поля и значения его констант приведены в табл. 25.

Таблица 25 – Тип, поле и константы для регистра *OE*

Регистр	<i>OE</i>
Тип	<i>PORT_OE_TypeDef</i>
Поле	<i>PORT_OE</i>
Константы	<i>PORT_OE_IN=0, PORT_OE_OUT=1</i>

Точно так же определяются остальные типы структуры *PORT_InitTypeDef*. Типы, имена полей и констант, их значения, а также РУП, для настройки которых они предназначены, приведены в таблицах с 26 по 31.

Для настройки режимов работы выводов порта следует создать в программе объект типа *PORT_InitTypeDef*, заполнить его поля и вызвать функ-

цию *PORT_Init()* с указателем на этот объект в качестве входного параметра. Например, в листинге 1 приведён программный код для настройки выводов портов *E* и *F*.

Таблица 26 – Типы, поля и константы для регистра *PULL*

Регистр	<i>PULL</i>	
Тип	<i>PORT_PULL_UP_TypeDef</i>	<i>PORT_PULL_DOWN_TypeDef</i>
Поле	<i>PORT_PULL_UP</i>	<i>PORT_PULL_DOWN</i>
Константы	<i>PORT_PULL_UP_OFF=0,</i> <i>PORT_PULL_UP_ON=1</i>	<i>PORT_PULL_DOWN_OFF=0,</i> <i>PORT_PULL_DOWN_ON=1</i>

Таблица 27 – Типы, поля и константы для регистра *PD*

Регистр	<i>PD</i>	
Тип	<i>PORT_PD_SHM_TypeDef</i>	<i>PORT_PD_TypeDef</i>
Поле	<i>PORT_PD_SHM</i>	<i>PORT_PD</i>
Константы	<i>PORT_PD_SHM_OFF = 0,</i> <i>PORT_PD_SHM_ON = 1</i>	<i>PORT_PD_DRIVER = 0,</i> <i>PORT_PD_OPEN = 1</i>

Таблица 28 – Тип, поле и константы для регистра *GFEN*

Регистр	<i>GFEN</i>
Тип	<i>PORT_GFEN_TypeDef</i>
Поле	<i>PORT_GFEN</i>
Константы	<i>PORT_GFEN_OFF = 0, PORT_GFEN_ON = 1</i>

Таблица 29 – Тип, поле и константы для регистра *FUNC*

Регистр	<i>FUNC</i>
Тип	<i>PORT_FUNC_TypeDef</i>
Поле	<i>PORT_FUNC</i>
Константы	<i>PORT_FUNC_PORT=0=00₂</i> - Режим ЦВВ <i>PORT_FUNC_MAIN = 1 = 01₂</i> - Основная функция <i>PORT_FUNC_ALTER=2=10₂</i> - Альтернативная функция <i>PORT_FUNC_OVERRID=3=11₂</i> - Переопределённая функция

Таблица 30 – Тип, поле и константы для регистра *PWR*

Регистр	<i>PWR</i>
Тип	<i>PORT_SPEED_TypeDef</i>
Поле	<i>PORT_SPEED</i>
Константы	<i>PORT_OUTPUT_OFF=0=00₂</i> - Передатчик отключён <i>PORT_SPEED_SLOW=1=01₂</i> - Длительность фронта 100 нс <i>PORT_SPEED_FAST=2=10₂</i> - Длительность фронта 20 нс <i>PORT_SPEED_MAXFAST=3=11₂</i> - Длительность фронта 10 нс

Таблица 31 – Тип, поле и константы для регистра *ANALOG*

Регистр	<i>ANALOG</i>
Тип	<i>PORT_MODE_TypeDef</i>
Поле	<i>PORT_MODE</i>
Константы	<i>PORT_MODE_ANALOG = 0, PORT_MODE_DIGITAL = 1</i>

Листинг 1.

```

// Создание объекта с именем PortInit типа
// PORT_InitTypeDef
static PORT_InitTypeDef PortInit;
// Настройка выводов PE6 и PE3 порта E на ввод цифровых
// данных.
// Заполнение полей
PortInit.PORT_Pin    = (PORT_Pin_6 | PORT_Pin_3);
PortInit.PORT_OE     = PORT_OE_IN; // табл. 25
PortInit.PORT_PULL_UP = PORT_PULL_UP_OFF; // табл. 26
PortInit.PORT_PULL_DOWN = PORT_PULL_DOWN_OFF;
PortInit.PORT_PD_SHM = PORT_PD_SHM_OFF; // табл.27
PortInit.PORT_PD     = PORT_PD_DRIVER; // табл. 27
PortInit.PORT_GFEN   = PORT_GFEN_OFF; // табл. 28
PortInit.PORT_FUNC   = PORT_FUNC_PORT; // табл. 29
PortInit.PORT_SPEED  = PORT_SPEED_SLOW; // табл. 30
PortInit.PORT_MODE   = PORT_MODE_DIGITAL; // табл. 31
PORT_Init(MDR_PORTE, &PortInit);
// Настройка выводов PF1 и PF0 порта F на вывод
// цифровых данных
PortInit.PORT_Pin    = (PORT_Pin_0 | PORT_Pin_1);
PortInit.PORT_OE     = PORT_OE_OUT; // табл. 25
PORT_Init(MDR_PORTF, &PortInit);

```

Функция *uint32_t PORT_ReadInputData (MDR_PORT_TypeDef * PORTx)* предназначена для считывания данных из регистра *RXTX* порта. Например, программный код:

```
unsigned int d;
```

```
d = PORT_ReadInputData (MDR_PORTA);
```

производит считывание значения из регистра *RXTX* порта *A* и сохраняет это значение в переменной *d*.

Функция *void PORT_Write(MDR_PORT_TypeDef * PORTx, uint32_t PortVal)* предназначена для записи значения *PortVal* в регистр *RXTX* порта. Например, вызов функции:

```
PORT_Write(MDR_PORTB, 0x0003);
```

записывает число 3 регистр *RXTX* порта *B*.

Функция *uint8_t PORT_ReadInputDataBit (MDR_PORT_TypeDef * PORTx, uint32_t PORT_Pin)* предназначена для считывания значения разрядов регистра *RXTX* порта. Маска *PORT_Pin* указывает разряды, значения которых необходимо считать. Функция возвращает 1, если хотя бы один из разрядов, установленных в маске, установлен в регистре *RXTX*, и 0, если все разряды сброшены. Например, программный код:

```
unsigned char d;
```

```
d = PORT_ReadInputData (MDR_PORTD, PORT_Pin_0);
```

производит чтение нулевого разряда регистра *RXTX* порта *D*.

Функция *void PORT_WriteBit (MDR_PORT_TypeDef * PORTx, uint32_t PORT_Pin, BitAction BitVal)* предназначена для записи значения *BitVal* в разряды регистра *RXTX*, установленные в маске *PORT_Pin*. Тип *BitAction* определён следующим образом:

```
typedef enum
```

```
{
```

```

Bit_RESET = 0,
Bit_SET
} BitAction;

```

Например, вызов функции

```
PORT_WriteBit (MDR_PORTF, PORT_Pin_0 | PORT_Pin_1, Bit_RESET);
```

производит сброс нулевого и первого разрядов регистра *RXTX* порта *F*.

Функция `void PORT_SetBits (MDR_PORT_TypeDef * PORTx, uint32_t PORT_Pin)` предназначена для установки разрядов регистра *RXTX*, которые установлены в маске *PORT_Pin*. Например, вызов функции

```
PORT_SetBits (MDR_PORTF, PORT_Pin_0 | PORT_Pin_1);
```

приведёт к установке нулевого и первого разрядов регистра *RXTX* порта *F*.

Функция `PORT_ResetBits()` предназначена для сброса разрядов регистра *RXTX*. Входные и выходные параметры функции такие же, как и у функции `PORT_SetBits()`. Например, вызов функции

```
PORT_ResetBits (MDR_PORTF, PORT_Pin_0 | PORT_Pin_1);
```

приведёт к сбросу нулевого и первого разрядов регистра *RXTX* порта *F*.

После АС подача тактовых сигналов на порты не производится с целью уменьшения энергопотребления. Для управления тактированием портов предназначена функция: `void RST_CLK_PCLKcmd (uint32_t RST_CLK_PCLK, FunctionalState NewState)`. Параметр *RST_CLK_PCLK* является маской и определяет порты, тактирование которых надо разрешить или запретить. Маска составляется посредством операции поразрядного логического ИЛИ констант формата *RST_CLK_PCLK_PORTX*, где вместо буквы *X* следует подставить букву имени порта. Параметр *NewState* может иметь два значения: *ENABLE* – разрешить тактирование и *DISABLE* – запретить тактирование портов. Например, вызов функции:

RST_CLK_PCLKcmd (RST_CLK_PCLK_PORTC | RST_CLK_PCLK_PORTD, ENABLE);

разрешает тактирование портов *C* и *D*. Определения констант формата *RST_CLK_PCLK_PORTX* приведены во включаемом файле "*MDR32F9Qx_rst_clk.h*".

Производить операции с отдельными разрядами РУП можно также с помощью непосредственного использования поразрядных логических операций. Именно они используются в программном коде функций. Например, сбросить 0 разряд регистра *RXTX* порта *F* можно двумя способами:

1) с помощью поразрядных логических операций

MDR_PORTF->RXTX &= ~ 0x01;

2) посредством вызова функции

PORT_ResetBits (MDR_PORTF, PORT_Pin_0);.

Из сравнения записей двух способов видно, что использование функций повышает читаемость программного кода и он становится более понятным.

Описание лабораторной установки

Лабораторной установкой является ДОП 1986EvBrd_48, описание которой приведено в [4]. В комплект поставки платы входит программатор, который является аналогом программатора *J-LINK*. Программатор следует подключать к разъёму *JTAG-A*. Электрическая принципиальная схема ДОП приведена в [5].

На ДОП два СИД, выполняющих функции ЕИ и обозначенных как *VD2* и *VD3* [5], подключены к выводам *PF0* и *PF1* порта *F* МК соответственно (сигнальные цепи *PF0_VD* и *PF1_VD*). Для программного управления СИД выводы *PF0* и *PF1* должны быть настроены на работу в режиме цифрового вывода. После этого для включения ЕИ надо в первый или нулевой разряды регистра *RXTX* записать единицу, а для выключения – нуль.

На ДОП установлены пять кнопок управления. Названия кнопок, а также выводов МК, к которым они подключены, приведены в табл. 32 [5]. Кнопки подключены по схеме, приведённой на рис. 4, в. При нажатии кнопок активным является низкий логический уровень. На электрической принципиальной схеме кнопка *SELECT* обозначена как *SEL*.

Таблица 32 – Название кнопок и выводов МК

Название кнопки	Вывод МК
<i>SELECT</i>	<i>PC0</i>
<i>UP</i>	<i>PD0</i>
<i>DOWN</i>	<i>PD1</i>
<i>LEFT</i>	<i>PE3</i>
<i>RIGHT</i>	<i>PE6</i>

2. Порядок проведения лабораторной работы

1. Предварительно создайте на жёстком диске компьютера новую рабочую папку, в которой будут находиться файлы создаваемой программы. Для выполнения лабораторной работы на персональном компьютере должна быть установлена среда разработки *Keil μ Vision*.

2. В соответствии со своим вариантом, полученным у преподавателя, разработайте алгоритм работы программы для МК. Варианты индивидуальных заданий приведены в табл. 33.

3. В соответствии с разработанным алгоритмом разработайте программу на языке программирования Си.

4. Произведите компиляцию разработанной программы и выполните процедуру программирования МК.

5. Убедитесь в правильности работы разработанной программы.

6. Составьте отчёт о выполненной лабораторной работе.

Буква Н в табл. 33 обозначает начало РЦВВ единичного индикатора, а буква К - соответственно конец этого режима. В этом режиме ЕИ включается и выключается через определённые интервалы времени (мигает). Обозначение «Вкл/Выкл» указывает, что после каждого нажатия кнопки указанный ЕИ должен изменить своё состояние на противоположное: если он включен, то после нажатия кнопки он должен выключиться и наоборот.

В качестве примера в листинге 2 приведена программа, которая выполняет вариант № 11 в табл. 33. Алгоритм работы программы приведён на рис. 5.

Таблица 33 - Варианты индивидуальных заданий

Вариант	Управление состоянием ЕИ VD2 и VD3 с помощью кнопок									
	<i>LEFT</i>		<i>RIGHT</i>		<i>UP</i>		<i>DOWN</i>		<i>SELECT</i>	
	VD2	VD3	VD2	VD3	VD2	VD3	VD2	VD3	VD2	VD3
1	Н		К		Вкл/ Выкл	Вкл		Выкл		
2	Вкл		Выкл		Н	Вкл		Выкл	К	
3	Вкл	Вкл	Выкл	Выкл		Н				К
4	Н		К		Вкл	Вкл	Выкл	Выкл		
5	Вкл/ Выкл			Вкл/ Выкл		Н				К
6	Вкл/ Выкл			Вкл/ Выкл	Н	Вкл	К			Выкл
7		Вкл/ Выкл	Вкл/ Выкл		Н	Вкл		Выкл	К	
8		Н		К	Вкл	Вкл	Выкл	Выкл		
9	Н		К		Вкл		Вкл		Выкл	Выкл
10		Н		К	Вкл		Выкл			Вкл/ Выкл
11	Вкл/ Выкл			Вкл/ Выкл	Н			Н	К	К

Листинг 2.

```
#include "MDR32F9Qx_port.h"
#include "MDR32F9Qx_rst_clk.h"
#define DELAY 10000

int main(void)
{
    static PORT_InitTypeDef PortInit;

    // Разрешение тактирования портов
    RST_CLK_PCLKcmd(RST_CLK_PCLK_PORTC | RST_CLK_PCLK_PORTD
    | RST_CLK_PCLK_PORTE | RST_CLK_PCLK_PORTF, ENABLE);
    // Настройка выводов для кнопок
    // Настройка выводов PD0 и PD1
    PortInit.PORT_Pin = (PORT_Pin_0 | PORT_Pin_1);
    PortInit.PORT_SPEED = PORT_SPEED_SLOW;
    PortInit.PORT_MODE = PORT_MODE_DIGITAL;
    PORT_Init(MDR_PORTD, &PortInit);
    // Настройка вывода PC0
    PortInit.PORT_Pin = (PORT_Pin_0);
    PORT_Init(MDR_PORTC, &PortInit);
    // Настройка выводов PE3 и PE6
    PortInit.PORT_Pin = (PORT_Pin_3 | PORT_Pin_6);
    PORT_Init(MDR_PORTE, &PortInit);

    // Настройка выводов PF0 и PF1 для управления СИД
    PortInit.PORT_Pin = (PORT_Pin_0 | PORT_Pin_1);
    PortInit.PORT_OE = PORT_OE_OUT;
    PORT_Init(MDR_PORTF, &PortInit);
```

```
while(1) // Основной рабочий цикл
{
    static unsigned cycle = 0;
    static unsigned char VD2 = 0, VD3 = 0;
    static unsigned BlinkVD2 = 0, BlinkVD3 = 0;
    // Обработка нажатий кнопок пользователем
    // Кнопка LEFT
    if(!PORT_ReadInputDataBit(MDR_PORTE,PORT_Pin_3))
    {
        while (!PORT_ReadInputDataBit(MDR_PORTE,
            PORT_Pin_3));
        BlinkVD2 = 0; VD2++;
    }
    // Кнопка RIGHT
    if(PORT_ReadInputDataBit(MDR_PORTE,PORT_Pin_6)==0)
    {
        while(!PORT_ReadInputDataBit(MDR_PORTE,
            PORT_Pin_6) );
        BlinkVD3 = 0; VD3++;
    }
    // Кнопка UP
    if(!PORT_ReadInputDataBit(MDR_PORTD,PORT_Pin_0))
    {
        while(!PORT_ReadInputDataBit(MDR_PORTD,
            PORT_Pin_0));
        BlinkVD2 = 1; cycle = 0; VD2++;
    }
    // Кнопка DOWN
    if(!PORT_ReadInputDataBit(MDR_PORTD,PORT_Pin_1))
    {
```

```

    while(!PORT_ReadInputDataBit(MDR_PORTD,
        PORT_Pin_1));
    BlinkVD3 = 1; cycle = 0; VD3++;
}
// Кнопка SELECT
if(!PORT_ReadInputDataBit(MDR_PORTC, PORT_Pin_0))
{
    while(!PORT_ReadInputDataBit(MDR_PORTC,
        PORT_Pin_0));
    BlinkVD2 = BlinkVD3 = 0;
}
// Определение необходимости переключения СИД
if(++cycle >= DELAY)
{
    cycle = 0;
    if(BlinkVD2)
        VD2++;
    if(BlinkVD3)
        VD3++;
}
// Управление СИД
if(VD2 & 0x01)
    PORT_SetBits(MDR_PORTF, PORT_Pin_0);
else
    PORT_ResetBits(MDR_PORTF, PORT_Pin_0);
if(VD3 & 1)
    PORT_SetBits(MDR_PORTF, PORT_Pin_1);
else
    PORT_ResetBits(MDR_PORTF, PORT_Pin_1);
}

```

}

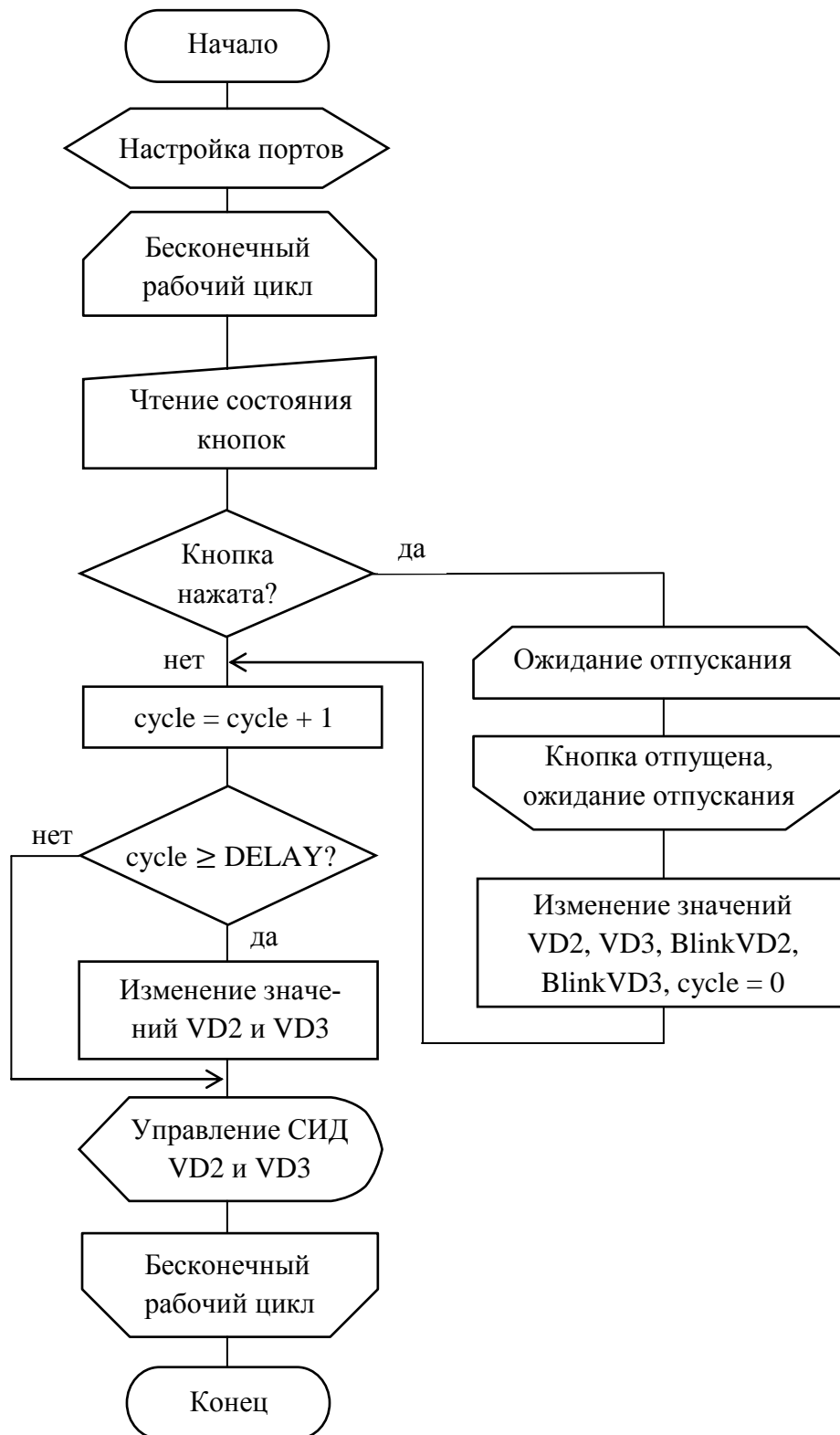


Рисунок 5 – Алгоритм работы программы.

При настройке режимов работы выводов МК для работы в режиме цифрового ввода с целью получения информации о состоянии кнопок не все поля объекта *PortInit* заполняются, потому что значения РУП после АС удовлетворяют требуемому режиму.

Константа *DELAY* задаёт количество рабочих циклов, через которое ЕИ должен изменить своё состояние на противоположное в РЦВВ. Переменная *cycle* предназначена для подсчёта количества выполненных рабочих циклов.

Нулевые разряды переменных *VD2* и *VD3* предназначены для хранения состояния одноимённых ЕИ: если нулевой разряд установлен, то ЕИ включен; иначе - выключен. Инкремент любой переменной приводит к инверсии её нулевого разряда.

Переменные *BlinkVD2* и *BlinkVD3* предназначены для указания работы единичных индикаторов *VD2* и *VD3* в РЦВВ. Если значение какой-нибудь переменной равно единице, то соответствующий ЕИ работает в РЦВВ.

Считывание состояния кнопки осуществляется функцией *PORT_ReadInputDataBit()*. При нажатии кнопки пользователь держит кнопку нажатой около 0,5 с. За это время основной рабочий цикл МК будет выполнен множество раз и при использовании только вышеназванной функции будет происходить многократное выполнение операторов в конструкции оператора условного перехода *if(!PORT_ReadInputDataBit())* на одно нажатие кнопки. Во избежание этого добавлен пустой цикл *while(!PORT_ReadInputDataBit());*, который предназначен для ожидания момента отпускания кнопки пользователем после её нажатия, в результате чего достигается однократное выполнения операторов, входящих в конструкцию оператора условного перехода *if(!PORT_ReadInputDataBit())*, после одного нажатия кнопки.

3. Оформление отчёта

Отчёт должен быть оформлен в соответствии с требованиями нормокон-троля и должен содержать:

- цель работы;
- задание на лабораторную работу;
- электрическую принципиальную схему лабораторной установки, кото-рая должна включать в себя только схемы подключения СИД, кнопок, разъё-мов для программирования и электрического питания МК;
- блок-схему алгоритма программы;
- текст программы на языке программирования Си;
- выводы о проделанной работе.

Контрольные вопросы

1. Что такое ПВВ и для чего он предназначен?
2. Приведите классификацию портов?
3. Какими функциональными возможностями обладает порты микрокон-троллера 1986ВЕ93У?
4. Какими функциями порта микроконтроллера 1986ВЕ93У можно управлять программно?
5. Как производится вычисление адреса РУП микроконтроллера 1986ВЕ93У?

Библиографический список

1. ГОСТ 25066-91. Индикаторы знакосинтезирующие. Термины, опреде-ления и буквенные обозначения [Текст]. – Взамен ГОСТ 25066-81: введ. 92-07-01. – М.: Стандартиформ, 2005. – 92 с.

2. Микросхема 32-разрядного однокристального микро-ЭВМ с памятью Flash-типа 1986BE9ху, К1986BE9ху, К1986BE92QI, К1986BE92QC, 1986BE91H4, К1986BE91H4, 1986BE94H4, К1986BE94H4 [Электронный ресурс]: Спецификация. – Версия 3.10.0 от 04.10.2016. – Электрон. дан. (7,54 Mbytes). – [Б.м.]: АО «ПКК Миландр», 2016. – Режим доступа: <http://ic.milandr.ru/upload/iblock/33e/33e6826dd444292426050ff6e39095a1.pdf>, свободный. – Загл. с экрана.

3. *MDR32F9Qx Standard Peripherals Library* [Электронный ресурс] . – Версия от 06.10.2015. – Электрон. дан. (34,58 Mbytes). – [Б.м.]: АО «ПКК Миландр», 2015. – Режим доступа: <http://ic.milandr.ru/upload/iblock/4c0/4c05fec706112f5b21f73eaf2e68fd51.rar>, свободный. – Загл. с экрана. – Яз. англ.

4. Демонстрационно-отладочная плата 1986EvBrd_48 [Электронный ресурс] : Техническое описание. – Версия 1.0 от 25.05.2010. – Электрон. дан. (1,05 Mbytes). – [Б.м.]: ЗАО «ПКК Миландр», 2010. – 1 электрон. опт. диск (CD-ROM). – Систем. требования: Pentium 100МГц ; 16 Мб RAM ; Windows 7 ; CD-ROM дисковод ; SVGA видеокарта, 256 цв. ; мышь. - Загл. с экрана. - CD-ROM входит в комплект поставки демонстрационно-отладочной платы 1986EvBrd_48.

5. Отладочная плата 1986BE93У [Электронный ресурс] : Схема электрическая принципиальная. – Revision 3. Последнее изменение 24.04.2014. – Электрон. граф. дан. (148 Kbytes). – [Б.м.]: ЗАО «ПКК Миландр», 2014. - 1 электрон. опт. диск (CD-ROM). – Систем. требования: Pentium 100МГц ; 16 Мб RAM ; Windows 7 ; CD-ROM дисковод ; SVGA видеокарта, 256 цв. ; мышь. – Загл. с экрана. - CD-ROM входит в комплект поставки демонстрационно-отладочной платы 1986EvBrd_48.

СОДЕРЖАНИЕ

1. Методические указания	4
2. Порядок проведения лабораторной работы	34
3. Оформление отчета	40
Библиографический список	41