

Министерство образования и науки  
Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра конструирования узлов и деталей радиоэлектронной аппаратуры  
(КУДР)

А.А. Бомбизов, А.Г. Лоцилов

РАБОТА С ПОРТАМИ ВВОДА-ВЫВОДА.  
ОРГАНИЗАЦИЯ ВВОДА ИНФОРМАЦИИ

Методические указания к выполнению  
лабораторной и самостоятельной работы  
по дисциплине «Программирование микроконтроллеров»

Томск 2017

## 1 Введение

В предыдущей работе была освоена среда программирования Arduino IDE и получены навыки организации вывода цифровых сигналов через порт ввода-вывода общего назначения.

Целью настоящей работы является освоение приемов программирования содержащихся в микроконтроллере портов ввода-вывода настроенных на вход.

## 2 Краткая теория

С внешним миром микроконтроллер взаимодействует через порты ввода-вывода [1, п. 14]. Структурная схема одного порта изображена на рисунке 1.

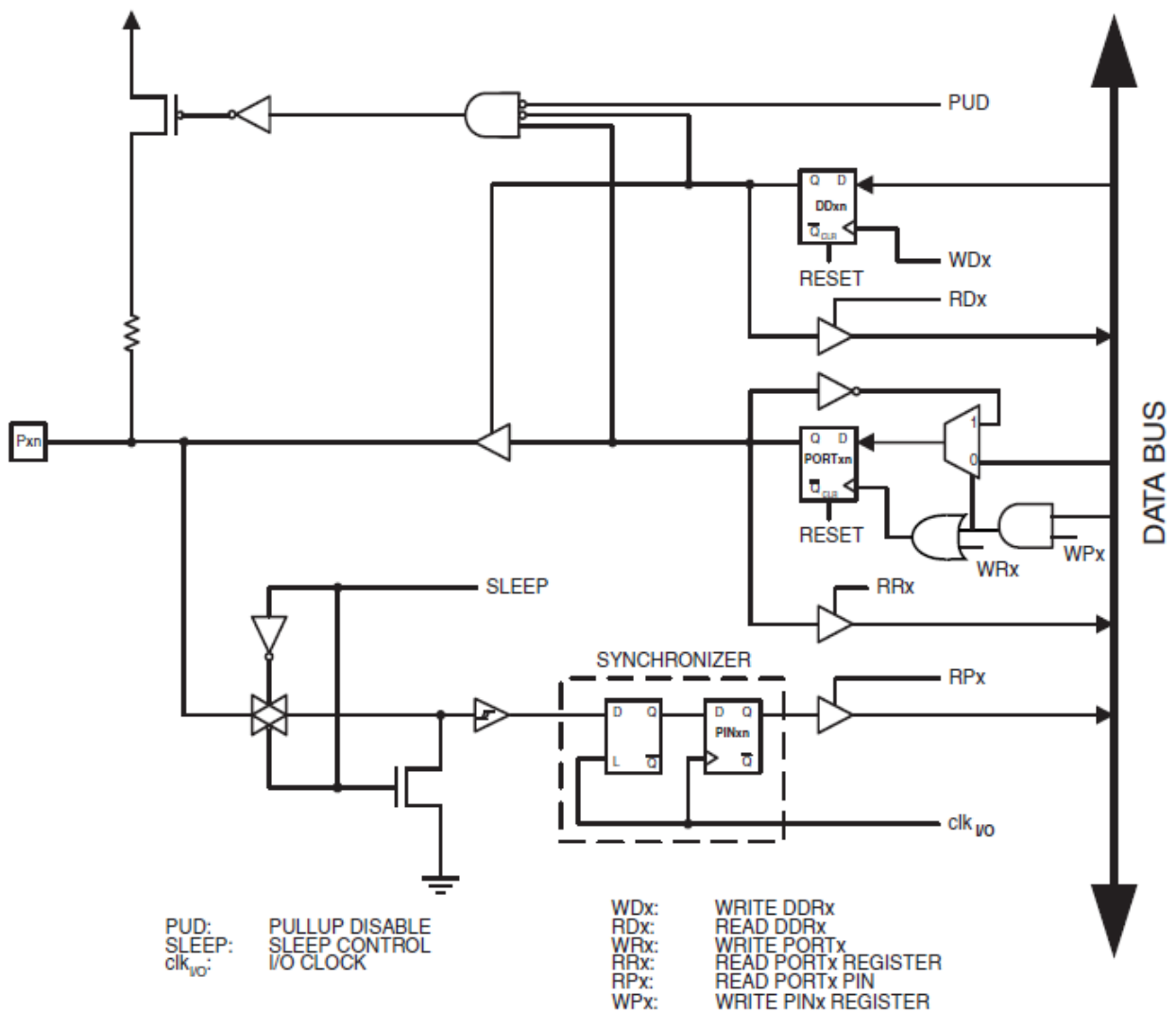


Рисунок 1 – Структурная схема одного вывода порта ввода-вывода микроконтроллера

Согласно рисунку 1 выходной контакт изображен в квадрате с подписью  $P_{xn}$ , где  $x$  – буква порта, а  $n$  – номер вывода в порте. За управление портами отвечают три регистра (регистр процессора, биты (флаги) которого отражают состояние процессора и свойства результатов предыдущих

операций): PORTx, PINx, DDRx. К примеру, за порт «В» отвечают три восьмиразрядных регистра PORTB, PINB, DDRB, а каждый бит в этих регистрах отвечает за соответствующий вывод порта. Это как линейка из восьми переключателей. DDRx, PORTx были рассмотрены в предыдущей работе.

PINx – регистр чтения. Из него можно только читать. В регистре PINx содержится информация о реальном текущем логическом уровне на выводах порта вне зависимости от настроек порта. Таким образом, для определения состояния вывода достаточно только прочесть значение соответствующего бита регистра.

Для упрощения понимания логики работы порта ввода-вывода удобней рассматривать схему замещения на рисунке 2.

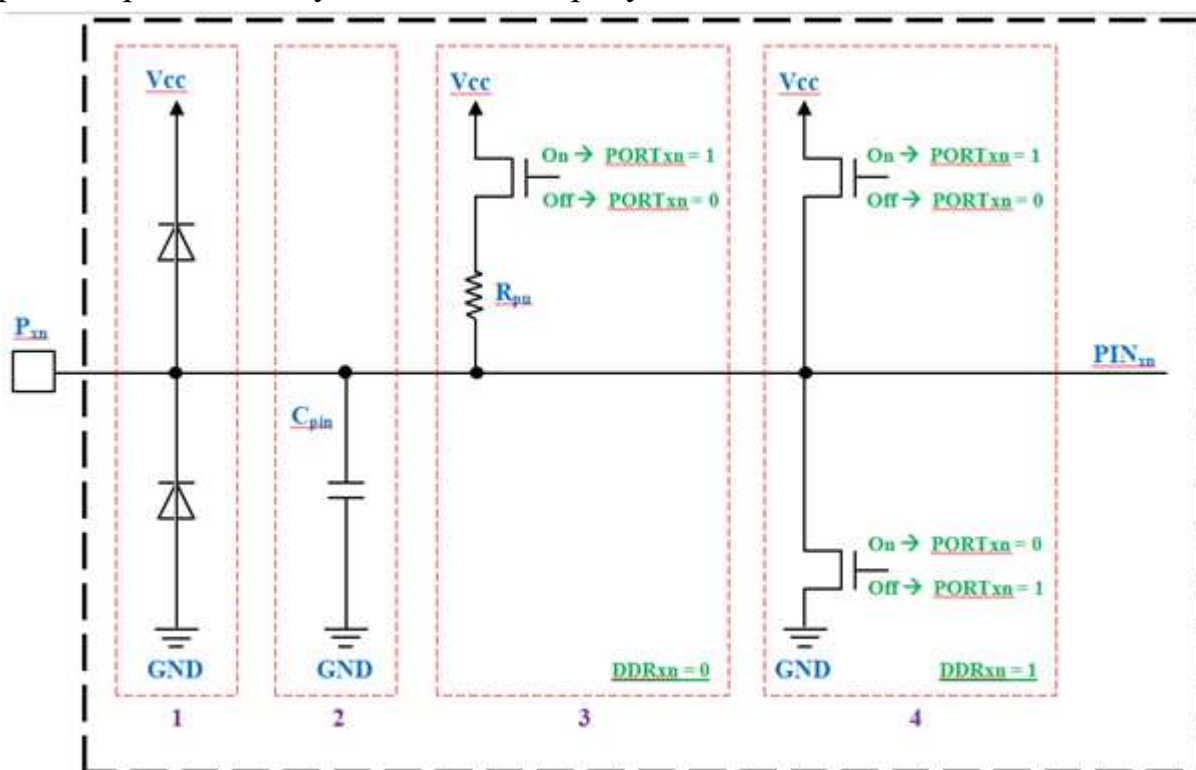


Рисунок 2 – Схема замещения порта ввода-вывода микроконтроллера

$P_{xn}$  – имя вывода порта микроконтроллера, где  $x$  буква порта (A, B, C или D),  $n$  номер разряда порта (7... 0);

$C_{pin}$  – паразитная емкость порта;

VCC – напряжение питания;

$R_{pu}$  – отключаемый нагрузочный резистор (pull-up);

$PORT_{xn}$  – бит  $n$  регистра PORTx;  $PIN_{xn}$  – бит  $n$  регистра PINx.

$DDR_{xn}$  – бит  $n$  регистра DDRx.

Всю схему замещения можно разделить на четыре блока:

1) Защита вывода микроконтроллера от кратковременных импульсов перенапряжения, выполненная с использованием двух диодов. Если

напряжения питания будет выше питания, то верхний диод откроется, и это напряжение будет отводиться на шину питания. Если на вывод попадет отрицательное (ниже нулевого уровня) напряжение, то оно будет нейтрализовано через нижний диод, подключенный на «землю».

2) Конденсатор – это паразитная емкость вывода. Из-за малых размеров она, как правило, не берется в расчет.

3) Блок включения подтягивающего резистора (pull-up), включенный между проводником, по которому распространяется электрический сигнал, и питанием, либо между проводником и землей. Подтягивающий резистор нужен, чтобы гарантировать на логическом входе, с которым соединен проводник, высокий (в первом случае) либо низкий (во втором случае) уровень в случаях:

- проводник не соединен с логическим выходом
- присоединённый логический выход находится в высокоимпедансном состоянии.

- когда разомкнут ключевой элемент на присоединённом логическом выходе, который устроен, как открытый вывод ключевого элемента. Ключевым элементом может быть полевой или биполярный транзистор, а открытым выводом — коллектор, эмиттер, сток или исток. А совсем в простом случае ключевым элементом может быть кнопка.

За подключение подтягивающего резистора согласно структурной схеме (рисунок 1) отвечают два регистра, управляющих логическим элементом «И». Это PORTxn и бит PUD в регистре MCUCR. По умолчанию бит PUD – Pull-up Disable – выставлен в логический ноль, поэтому для активации подтягивающего резистора никаких дополнительных манипуляций с этим битом не требуется выполнять.

4) Блок управления выходом. Эта часть работы была рассмотрена в предыдущей работе.

Для выполнения операций с помощью средств Arduino IDE с выводами порта, настроенными на выход, необходимо использовать функции *pinMode()*, *digitalRead()*.

Далее представлен пример использования данных функций

```
int led1=13;
//A1 – константа, соответствующая в среде Arduino IDE кнопке S1
int button1=A1;
int val1 = 0;
void setup()
{
    pinMode(led1, OUTPUT); // настройка led1 на выход
```

```

    pinMode(button1, INPUT); // настройка button1 на вход
}
void loop()
{
    // выполнение чтения в переменную val1 состояния кнопки S1
    val1 = digitalRead(button1);
    digitalWrite(led1, val1); // управление светодиодом D1
    delay(250); // задержка 250 мс
}

```

На практике достаточно часто работа с микроконтроллером осуществляется путём прямого обращения к регистрам. Физическое изменение состояния на входе микроконтроллера отражается в изменении состояния соответствующего бита в регистре PINx. Битовая схема регистра PINB представлена на рисунке 3.

Bit	7	6	5	4	3	2	1	0
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Рисунок 3 – Битовая схема регистра

Согласно битовой схеме каждый бит с наименованиями PINB0–PINB7 соответствуют выводам микроконтроллера PB0–PB7, расположение можно рассмотреть на рисунке 4.

Как было описано в прошлой работе, переключение вывода на выход или на вход осуществляется путём установки соответствующего бита в регистре в 1 или 0, соответственно. Например, для переключения вывода PB4 на выход необходимо выполнить следующую операцию:

$$DDRB |= (1 << 4);$$

то есть выполнение операции побитового ИЛИ значения в регистре с маской BIN:00010000.

Переключение вывода PB4 на вход достаточно выполнить операцию:

$$DDRB \&= \sim(1 << 4);$$

где & – операция побитового И, ~ – инверсия, которая приводит маску к виду BIN:11101111.

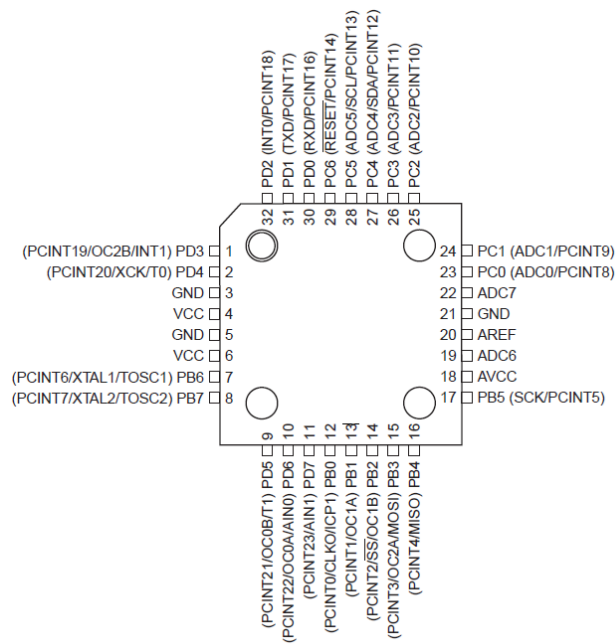


Рисунок 4 – Расположение выводов (цоколёвка – pinout)

В программе определение значения бита регистре может быть выполнена двумя способами:

1)  $val1 = (PINB \& (1 \ll 4)) \gg 4;$

где  $val1$  – предварительно объявленная целочисленная переменная. А действия справа от « $\ll$ » трактуются следующим образом:

а) рассчитывается маска ( $1 \ll 4$ ), в результате чего получится число вида BIN:00010000;

б) выполняется операция побитового И со значением регистра PINB: BIN:XXXXXXXX&00010000=000X0000. В результате будет получено число, четвертый бит которого будет соответствовать значению на выводе микроконтроллера;

в) выполнение сдвига вправо на 4. BIN: 000X0000  $\gg$  4 = 0000000X. В результате последней операции будет получено число 0 или 1, которое в дальнейшем может быть использовано в программе.

2)

```

if((PINB&(1<<4)))
{
//то на входе единица
}
else
{
//иначе на входе ноль
}

```

### 3 Порядок выполнения работы

В практической части работы должны быть решены две задачи: освоение работы с портами ввода-вывода, настроенными на режим входа; закрепление полученных навыков при решении задачи счетчика (значение счетчика должно отображаться в двоичной форме на светодиодах; увеличение и уменьшение значения выполняется кнопками S1–S2; сброс счетчика выполняется кнопкой S3). Выполнение работы должно быть выполнено в двух вариантах: управление режимами и состояниями выводов путём непосредственного обращения к регистрам микроконтроллера; управление с помощью функций Arduino IDE. Для этого необходимо выполнить следующие действия:

3.1 Изучите предложенный в п. 2 теоретический материал.

3.2 Подключите отладочную плату к компьютеру как описано в теоретической части.

3.3 Создайте новый проект (меню Файл->Новый) и сохраните его в свой рабочий каталог.

Освоение работы с портами ввода-вывода, настроенными на режим входа:

3.4 С использованием функций Arduino IDE реализуйте алгоритм переключения светодиода по нажатию кнопки.

3.5 Реализуйте алгоритмы, выполненные в пункте 3.4, но только переключение светодиода должно происходить **однократно** по отпусканию кнопки или по нажатию.

3.6 Заведите в программе переменную для хранения значения 4х-битного счетчика.

3.7 Реализуйте алгоритм для отображения значения счетчика на светодиодах отладочного комплекта.

3.8 Реализуйте алгоритмы увеличения, уменьшения и сброса значения счетчика при нажатии соответствующих кнопок.

3.9 Определите по [2] наименования выводов микроконтроллера, соответствующих кнопкам S1–S3.

3.10 Создайте новый проект и реализуйте алгоритмы, выполненные в пунктах 3.4 – 3.8, но только с использованием прямого обращения к регистрам микроконтроллера.

3.11 Оформите отчет, содержащий титульный лист и разделы: введение, ход выполнения работы, ответы на контрольные вопросы и выводы.

3.12 Защитите отчет у преподавателя.

#### **4 Контрольные вопросы**

- 4.1 Каким образом переключить вывод микроконтроллера на вход?
- 4.2 Какого назначения подтягивающего резистора?
- 4.3 Сколько выводов у микроконтроллера, с которыми можно работать как с портами ввода вывода?
- 4.4 Какой регистр отвечает за состояние на выводе микроконтроллера?

#### **Список литературы**

1. ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES, IN-SYSTEM PROGRAMMABLE FLASH.– URL: [http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Complete.pdf](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf) (дата обращения: 10.01.2017).
2. UNO Schematic – Arduino.– URL: <https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf> (дата обращения: 10.01.2017).