

1. Как расшифровывается аббревиатура GPIO?

Порт ввода/вывода общего назначения (General purpose input/output - GPIO) является программно-управляемым элементом микроконтроллера. На аппаратном уровне порт объединяет несколько выводов микроконтроллера в группу, работа которой программируется общим набором регистров.

2. Какова разрядность портов в микроконтроллерах STM32F? Поясните.

В STM32F порты имеют 16 выводов и являются 16-ти разрядными, т.е. с помощью управляющего регистра можно одной командой микроконтроллера изменить сигналы на всех выводах одного порта.

3. Как обозначаются выводы GPIO на принципиальной схеме?

Обычно названия портов обозначаются латинскими буквами A, B, C и т.д. Выводы (линии) порта нумеруются цифрами 0, 1, 2 и т.д. Причем нуль используется для обозначения вывода, который соответствует самой младшей группе бит управляющего регистра, и далее по старшинству. Так 4-ый вывод порта B на схеме микроконтроллера обозначают – PB4. Аналогичным образом обозначают остальные выводы портов ввода/вывода общего назначения

4. Каким образом осуществляется доступ к GPIO в программе?

За каждый порт GPIO отвечает несколько регистров. Однако перед началом настройки управляющих регистров порта нужно включить их тактирование, разрешив передачу сигнала системного таймера в блок порта.

5. Почему по умолчанию тактирование портов GPIO отключено?

В технологии КМОП потребление энергии происходит только при переключении состояния. Без тактирования нет переключений. Поэтому для экономии энергии после перезагрузки микроконтроллера тактирование его периферии отключено.

6. Какой регистр отвечает за управление тактированием GPIO? Как

осуществляется доступ к нему из программы?

За управление тактированием элементов микроконтроллера отвечает система регистров RCC. Для управления тактированием периферии используется регистр RCC_AHBENR (программный доступ: RCC->AHBENR)

7. Как включить тактирование порта C, используя численное значение?

RCC->AHBENR |= RCC_AHBENR_GPIOCEN

8. Как отключить тактирование порта B, не затронув управления тактированием других GPIO и блоков микроконтроллера?

А для выключения тактирования только порта B нужно указать команду:

RCC->AHBENR &= ~RCC_AHBENR_GPIOBEN;

9. Какие регистры используются для конфигурирования линий порта?

Каждая линия порта GPIO имеет непростую систему управления, которая позволяет определять направление потока данных при реализации основной функции, а также переключаться на альтернативную функцию (подключать к выводу другой блок микроконтроллера) или включать аналоговый режим работы (вывод обеспечивает передачу сигнала до/от АЦП, ЦАП, компаратора). Таким образом, для управления каждый порт ввода-вывода общего назначения имеет четыре 32-битных регистра конфигурации (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR и GPIOx_PUPDR)

10. Управляющие регистры порта позволяют задавать единую конфигурацию для всех линий порта или отдельно конфигурировать каждую линию?

Порт ввода/вывода общего назначения (General purpose input/output - GPIO) является программно-управляемым элементом микроконтроллера. На аппаратном уровне порт объединяет несколько выводов микроконтроллера в группу, работа которой программируется общим набором регистров. В STM32F порты имеют 16 выводов и являются 16-ти разрядными, т.е. с помощью управляющего регистра можно одной командой микроконтроллера

изменить сигналы на всех выводах одного порта. **Отдельно конфигурировать каждую линию.**

11. Каково назначение регистра GPIOx_MODER?

Регистр GPIOx_MODER (программный доступ: GPIOx->MODER) (x – буква порта, изменяется от A до F) используется для переключения между вводом/выводом бинарных данных (основная функция), альтернативной функцией и аналоговым режимом работы линий. Линии порта в регистре обозначаются MODERx, где x – номер линии порта, который изменяется от 0 до 15. Для настройки режима работы каждой линии порта используется два бита, определяющие 4 возможных комбинации:

00 – режим ввода бинарных данных (основная функция);

01 – режим вывода бинарных данных (основная функция);

10 – режим альтернативной функции;

11 – аналоговый режим работы линии.

Все биты регистра GPIOx_MODER можно как читать в программе (r), так и устанавливать новые значения (w). Например, для включения линии 3 порта B в режим альтернативной функции следует указать команду:

```
GPIOB->MODER |= 0x00000080;
```

или

```
GPIOB->MODER |= GPIO_MODER_MODER3_1; //Используя специальную константу
```

Переключить линию 3 порта A в режим ввода бинарных данных можно командой:

```
GPIOA->MODER &= ~0x000000C0;
```

или

```
GPIOA->MODER &= ~GPIO_MODER_MODER3; //Используя специальную константу
```

12. Можно ли с помощью регистра GPIOx_MODER управлять подтягивающими резисторами линии порта?

Нет. Регистр GPIOx_PUPDR (программный доступ: GPIOx->PUPDR) (x – буква порта, изменяется от А до F) используется для подключения/отключения подтягивающих резисторов к выводу микроконтроллера.

13. Каково назначение регистра GPIOx_OTYPER? Поясните.

Регистр GPIOx_OTYPER (программный доступ: GPIOx->OTYPER) (x – буква порта, изменяется от А до F) используется для указания режима вывода бинарных данных.

Линии порта в регистре обозначаются OTx, где x – номер линии порта, который изменяется от 0 до 15. Для настройки режима работы каждой линии порта используется один бита, определяющий 2 состояния:

0 – вывод данных в режиме «push-pull» (PP)(Обычный активный выход. При низком логическом уровне напряжение на выводе равно 0, при высоком – напряжение близко к напряжению питания микроконтроллера, обычно +3В. Режим используется по умолчанию после перезагрузки микроконтроллера);

1 – вывод данных в режиме «open-drain»(OD)(Работа выхода в режиме «открытый сток»).

Все биты регистра GPIOx_OTYPER можно как читать в программе (r), так и устанавливать новые значения (w). Например, для включения линии 2 порта В в режим OD следует указать команду:

```
GPIOB->OTYPER |= 0x00000004;
```

или

```
GPIOB->MODER |= GPIO_OTYPER_OT2; //Используя специальную константу
```

Переключиться обратно в режим PP можно командой:

```
GPIOB->OTYPER &=~0x00000004;
```

или

```
GPIOB->OTYPER &=~GPIO_OTYPER_OT2; //Используя специальную константу
```

14. Какие регистры используются для хранения входных и выходных данных порта? Как осуществляется к ним доступ из программы?

15. Каким образом можно управлять обновлением данных в выводных и входных регистрах порта? Приведите пример изменения режима обновления данных.

Регистр GPIOx_ODR (программный доступ: GPIOx->ODR) (x – буква порта, изменяется от А до F) используется для вывода (параллельного) данных на линии порта, работающих в режиме «на выход» (рис. 9). Линии порта в регистре обозначаются ODRx, где x – номер линии порта, который изменяется от 0 до 15.

Все биты регистра GPIOx_ODR можно как читать в программе (r), так и устанавливать новые значения (w). Операции чтения и записи в этот регистр не являются атомарными, т.е. они могут быть приостановлены в процессе обработки микроконтроллером прерывания. Пример: обновления данных на выводах порта С можно добиться командой:

```
uint32_t Data;  
Data=0xAC002F19;  
GPIOC->ODR=Data;
```

16. Что произойдет при выполнении команды: GRIOA->PUPDR=0x01000020;

17. Что произойдет при выполнении команды: GPIOB->IDR=0x0040f080;

18. В какой регистр попадают данные, если вывод порта работает на вход?

Регистр GPIOx_IDR (программный доступ: GPIOx->IDR) (x – буква порта, изменяется от А до F) используется для ввода (параллельного) данных с линий порта, работающих в режиме «на вход» (рис. 8). Линии порта в регистре обозначаются IDRx, где x – номер линии порта, который изменяется от 0 до 15.

Биты регистра GPIOx_IDR можно только читать в программе (r). Например, для ввода данных со всех линий порта В можно использовать такой код:

```
uint32_t Data;  
Data=GPIOB->IDR
```

19. Изменяться ли данные регистра GPIOB->IDR, если в регистр GPIOB->ODR

записать новое значение?

Регистр GPIOx_ODR (программный доступ: GPIOx->ODR) (x – буква порта, изменяется от A до F) используется для вывода (параллельного) данных на линии порта, работающих в режиме «на выход» (рис. 9). Линии порта в регистре обозначаются ODRx, где x – номер линии порта, который изменяется от 0 до 15.

Все биты регистра GPIOx_ODR можно как читать в программе (r), так и устанавливать новые значения (w). Операции чтения и записи в этот регистр не являются атомарными, т.е. они могут быть приостановлены в процессе обработки микроконтроллером прерывания. Пример: обновления данных на выводах порта C можно добиться командой:

```
uint32_t Data;  
  
Data=0xAC002F19;  
  
GPIOC->ODR=Data;
```

Да, возможно.

20. Каково назначение регистра GPIOx_BSRR? Поясните.

Регистр GPIOx_BSRR (программный доступ: GPIOx->BSRR) (x – буква порта, изменяется от A до F) используется для установки и сброса отдельных бит регистра GPIOx_ODR с помощью атомарной операции записи. Регистр разделен на две части. Биты с 0 по 15 используются для установки в «1» битов регистра GPIOx_ODR. При этом соответствующие биты регистра GPIOx->BSRR обозначаются BSx, где x – номер линии порта, который изменяется от 0 до 15.

Биты с 16 по 31 используются для сброса в «0» битов регистра GPIOx_ODR. При этом соответствующие биты регистра GPIOx->BSRR обозначаются BRx, где x – номер линии порта, который изменяется от 0 до 15.

Регистр можно использовать только на запись значений (w). Например, для установки «1» на 12 линии порта B, работающей в режиме «на выход», нужно выполнить команду:

```
GPIOB->BSRR |= 0x00001000;
```

А сброс в ноль сигнала на этой линии выполнит команда:

```
GPIOB->BSRR |= 0x10000000;
```

В команде:

```
GPIOB->BSRR |= 0x10001000;
```

приоритет будет отдан «установке» и после нее на 12-ом выводе порта B зафиксирован «1».

21. На вывода порта B необходимо установить значение 0xE46A. Как можно выполнить эту операцию с помощью регистра GPIOx_BSRR?

```
GPIOB->BSRR |= 1110010001101010
```

22. В чем различие установки выходных данных с помощью регистров GPIOB->ODR и GPIOx_BSRR?

Регистр GPIOx_ODR (программный доступ: GPIOx->ODR) (x – буква порта, изменяется от A до F) используется для вывода (параллельного) данных на линии порта, работающих в режиме «на выход» (рис. 9). Линии порта в регистре обозначаются ODRx, где x – номер линии порта, который изменяется от 0 до 15.

Все биты регистра GPIOx_ODR можно как читать в программе (r), так и устанавливать новые значения (w). Операции чтения и записи в этот регистр не являются атомарными, т.е. они могут быть приостановлены в процессе обработки микроконтроллером прерывания. Пример: обновления данных на выводах порта C можно добиться командой:

```
uint32_t Data;
```

```
Data=0xAC002F19;
```

```
GPIOC->ODR=Data;
```

Регистр GPIOx_BSRR (программный доступ: GPIOx->BSRR) (x – буква порта, изменяется от A до F) используется для установки и сброса отдельных бит регистра GPIOx_ODR с помощью атомарной операции записи. Регистр разделен на две части. Биты с 0 по 15 используются для установки в «1» битов регистра GPIOx_ODR. При этом соответствующие биты регистра GPIOx->BSRR обозначаются BSx, где x – номер линии порта, который изменяется от 0 до 15.

Биты с 16 по 31 используются для сброса в «0» битов регистра GPIOx_ODR. При этом соответствующие биты регистра GPIOx->BSRR обозначаются BRx, где x – номер линии порта, который изменяется от 0 до 15.

Регистр можно использовать только на запись значений (w). Например, для установки «1» на 12 линии порта B, работающей в режиме «на выход», нужно выполнить команду:

```
GPIOB->BSRR |= 0x00001000;
```

А сброс в ноль сигнала на этой линии выполнит команда:

```
GPIOB->BSRR |= 0x10000000;
```

В команде:

```
GPIOB->BSRR |= 0x10001000;
```

приоритет будет отдан «установке» и после нее на 12-ом выводе порта B зафиксируется «1».

23. Чем отличаются регистры GPIOx_BSRR и GPIOx_BRR?

Регистр GPIOx_BSRR (программный доступ: GPIOx->BSRR) (x – буква порта, изменяется от A до F) используется для установки и сброса отдельных бит регистра GPIOx_ODR с помощью атомарной операции записи. Регистр разделен на две части. Биты с 0 по 15 используются для установки в «1» битов регистра GPIOx_ODR. При этом соответствующие биты регистра GPIOx->BSRR обозначаются BSx, где x – номер линии порта, который изменяется от 0 до 15.

Биты с 16 по 31 используются для сброса в «0» битов регистра GPIOx_ODR. При этом соответствующие биты регистра GPIOx->BSRR обозначаются BRx, где x – номер линии порта, который изменяется от 0 до 15.

Регистр можно использовать только на запись значений (w). Например, для установки «1» на 12 линии порта B, работающей в режиме «на выход», нужно выполнить команду:

```
GPIOB->BSRR |= 0x00001000;
```

А сброс в ноль сигнала на этой линии выполнит команда:

```
GPIOB->BSRR |= 0x10000000;
```

В команде:

GPIOB->BSRR |= 0x10001000;

приоритет будет отдан «установке» и после нее на 12-ом выводе порта B зафиксируется «1».

Регистр GPIOx_BRR (программный доступ: GPIOx->BRR) (x – буква порта, изменяется от A до F) используется для сброса отдельных бит регистра GPIOx_ODR с помощью атомарной операции записи. Линии порта в регистре обозначаются BRx, где x – номер линии порта, который изменяется от 0 до 15.

Регистр можно использовать только на запись значений (w). Работа с ним аналогична работе с регистром GPIOx_BSRR за исключением возможности установки бит регистра GPIOx_ODR.

24. Какой из регистров лучше использовать, когда требуется регулярное обновление данных на выхода порта?

Регистр GPIOx_ODR (программный доступ: GPIOx->ODR) (x – буква порта, изменяется от A до F) используется для вывода (параллельного) данных на линии порта, работающих в режиме «на выход» (рис. 9). Линии порта в регистре обозначаются ODRx, где x – номер линии порта, который изменяется от 0 до 15.

25. Каково назначение регистра GPIOx_AFRL? Поясните.

Регистр GPIOx_AFRL (программный доступ: GPIOx->AFRL) (x – буква порта, изменяется от A до F) используется для выбора одной из 8 альтернативных функций на выводе порта для линий с 0 по 7.

Для настройки режима работы каждой линии порта используется 4 бита, определяющие 16 возможных комбинации, но для выбора доступно только 8 альтернативных функций, поэтому сочетания 1xxx зарезервированы и не используются.

Регистр GPIOx_AFRH (программный доступ: GPIOx->AFRH) (x – буква порта, изменяется от A до F) используется для выбора одной из 8 альтернативных функций на выводе порта для линий с 9 по 15.

Для настройки режима работы каждой линии порта используется 4 бита, определяющие 16 возможных комбинации, но для выбора доступно только 8

альтернативных функций, поэтому сочетания 1xxx зарезервированы и не используются.

26. О каких альтернативных функциях линий GPIO Вам известно?

Каждая линия порта GPIO имеет простую систему управления, которая позволяет определять направление потока данных при реализации основной функции, а также переключаться на альтернативную функцию (подключать к выводу другой блок микроконтроллера) или включать аналоговый режим работы (вывод обеспечивает передачу сигнала до/от АЦП, ЦАП, компаратора).

Таким образом, для управления каждый порт ввода-вывода общего назначения имеет четыре 32-битных регистра конфигурации (**GPIOx_MODER**, **GPIOx_OTYPER**, **GPIOx_OSPEEDR** и **GPIOx_PUPDR**), два 32-битных регистра данных (**GPIOx_IDR** и **GPIOx_ODR**) и 32-битный регистр установки / сброса (**GPIOx_BSRR**). Порты A и B также имеют 32-битный регистр блокировки (**GPIOx_LCKR**) и два 32-битных регистра выбора альтернативных функций (**GPIOx_AFRH** и **GPIOx_AFRL**).

27. Какие альтернативные функции имеет PB.10?

28. Какая альтернативная функция соответствует AF4 линии PB.8?

29. Для чего предназначен регистр GPIOx_LCKR?

Регистр **GPIOx_LCKR** (программный доступ: **GPIOx->LCKR**) (x – буква порта, изменяется от A до F) используется для блокировки конфигурации соответствующих линий микроконтроллера.

Порты A и B также имеют 32-битный регистр блокировки (**GPIOx_LCKR**).

30. Можно ли изменить содержимое регистра **GPIOB->ODR**, если включена блокировка конфигурирования всех линий порта B? Поясните.