

Цель работы: Написать программное обеспечение для микроконтроллера stm32f407vg, включающее и выключающее светодиоды на плате stm32f4-discovery

Часть 1: Порты общего назначения.

Общее описание порта ввода-вывода.

Порт ввода вывода является программно управляемым элементом микроконтроллера. Порт состоит из отдельных выводов, являющихся ножками микроконтроллера. В stm32f порты имеют 16 выводов. Такие порты по числу выводов называются 16-ти разрядными. Обычно названия портов обозначаются латинскими буквами A, B, C и т.д. А выводы нумеруются цифрами 0, 1, 2 и т.д. Причём нулём нумеруется самый младший бит и далее по старшинству.

Выводы микроконтроллера могут работать в двух режимах: на вход и на выход. В первом случае с ножки снимается текущее напряжение и это значение можно использовать в программе. Например, это может быть сигнал готовности какого-то устройства. Т.е. мы ждем, когда на определённой ножке установится 1. Когда это произошло, мы можем работать дальше, т.к. наш гипотетический прибор готов к взаимодействию с МК. Во втором случае программист задаёт будет ли на выводе логический ноль или логическая единица. Этот режим может использоваться для управления внешними устройствами. Например, чтобы включить какой-то прибор требуется установить логическую единицу на определённом выводе микроконтроллера.

Отметим, что все выводы микроконтроллера могут работать независимо друг от друга. Т.е. один порт может одновременно управлять несколькими устройствами, присоединёнными к разным выводам этого порта. Т.е. один и тот же порт может одновременно установить логическую единицу на один из своих выводов и затем ждать прихода единицы, означающей готовность устройства, на другой вывод.

Также к выводам общего назначения могут подключаться внутренние устройства МК в случае, если им надо выводить наружу какие то сигналы. При этом порт общего назначения может быть запрограммирован для использования в режиме альтернативной функции. В этом режиме вывод перестаёт работать как вывод общего назначения и подключается к заданному внутреннему устройству МК, которое через этот порт сможет передавать данные. Очевидно, что альтернативная функция может быть

запрограммирована как при работе на вход, так и на выход. В первом случае есть особенности, которые будут рассмотрены в следующей главе.

Принцип работы порта ввода-вывода stm32f

Блок-схема порта ввода вывода микроконтроллера семейства stm32f представлена на рис 1.1

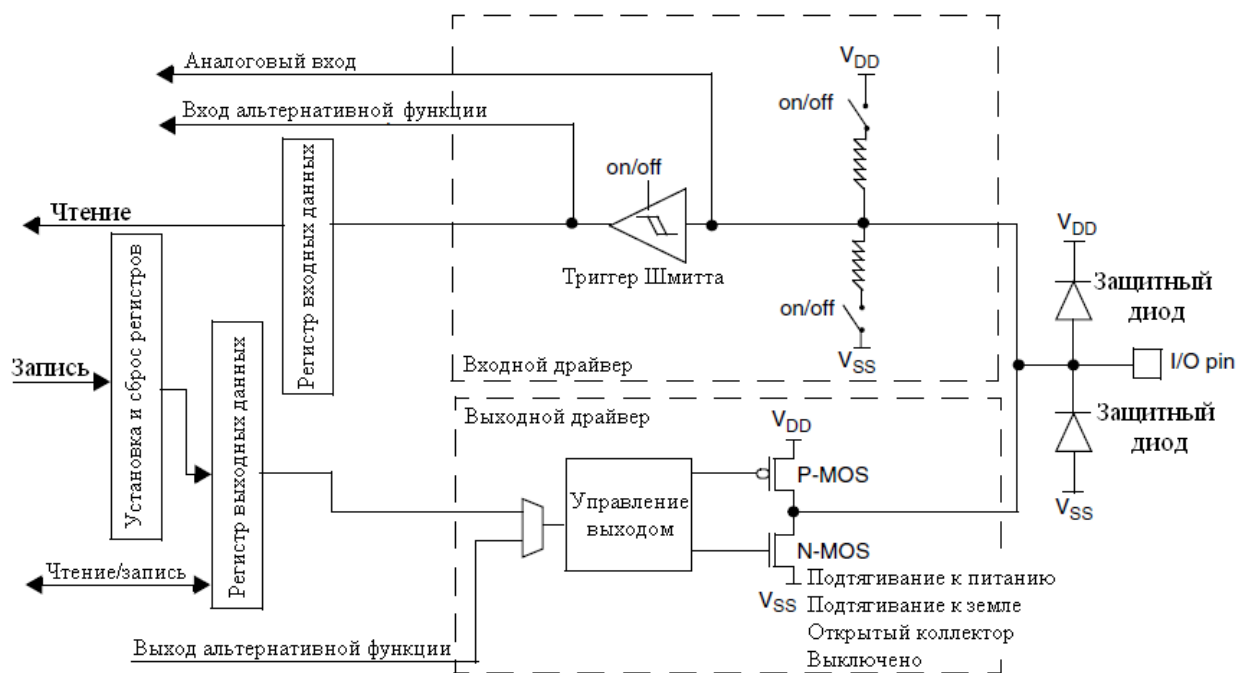


Рис. 1.1

Порт ввода-вывода микроконтроллера работает либо на вход, либо и на выход. Одновременно работать в обоих направлениях выводы микроконтроллера не могут. Поэтому всю структуру порта ввода-вывода можно разделить на два больших блока: блок управления входом и блок управления выходом.

Общими для входа и выхода являются два защитных диода, ограничивающих напряжение, которое может быть подано на вывод микроконтроллера. На принцип работы порта при нормальных условиях работы (напряжение на выводе не превышает напряжения питания МК) эти диоды никак не влияют.

В блоке приёмной части два резистора отвечают за подтяжку вывода к питанию или к земле. Это может пригодиться, если предполагается, что входной сигнал может оказаться «висящим в воздухе», т.е. быть никуда не подключенным. Кроме того, вход можно никуда не подтягивать.

Далее идёт разделение в зависимости от настроек. Сигнал может пойти на альтернативную функцию (например USART) или на регистр входных значений. В данном регистре записывается состояние порта при работе его на вход. Узнать о состоянии порта можно, прочитав значение из этого регистра.

В итоге вывод, при работе на выход может быть запрограммирован на одно из четырёх состояний: подтянутый к питанию, подтянутый к земле, «висящий в воздухе» и вход аналогового сигнала. В последнем случае порт подключается ко входу аналогово-цифрового преобразователя и не зависит от подтягивающих резисторов. Для работы альтернативной функции вход всё равно надо установить в одно из трёх первых состояний.

При работе на выход данные записываются в выходной регистр с помощью регистра установки/сброса бита. Мультиплексор подключает либо регистр порта общего назначения, либо выход альтернативной функции к блоку управления выходом. Этот блок с помощью КМОП схемы из двух транзисторов устанавливает на выходе требуемое значение.

Выход может быть в трёх состояниях: равный единице, равный нулю и выход альтернативной функции.

Часть 2: Алгоритм выполнения лабораторной работы

Настройка портов ввода-вывода

Определение подключения светодиодов

Откройте файл «Схема.pdf». Найдите там изображения светодиодов (LED) (стр. 6). Определите, к каким выводам микроконтроллера подключены эти светодиоды.

Примечание: Порты микроконтроллера называются буквами по порядку латинского алфавита (A, B, C ...). Каждый порт включает в себя 16 выводов, нумеруемых арабскими цифрами. Т.е. 12 вывод порта D будет полностью называться PORTD 12. При этом на схеме скорее всего будет обозначен как PD12.

Включение тактирования

Вся внутренняя периферия микроконтроллера по-умолчанию не имеет тактирования и, как следствие, не работает. Чтобы она заработала, её надо затактировать, т.е. подать на неё тактовый сигнал. Для включения тактирования порта надо написать строку:

```
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOx, ENABLE);
```

Где x – буква, обозначающая порт.

Настройка порта ввода-вывода

Для настройки порта ввода-вывода надо заполнить структуру. Определяется структура следующим образом:

```
GPIO_InitTypeDef  PORT_init_struct;
```

Где GPIO_InitTypeDef – тип структуры. PORT_init_struct – название. Название может быть произвольным с некоторыми ограничениями.

Примечание: Все переменные определяются либо вне функции, либо в самом её начале.

Структура имеет следующие параметры:

PORT_init_struct.GPIO_Pin – Определяет номер программируемого вывода.

Записывается так:

```
PORT_init_struct.GPIO_Pin = GPIO_Pin_x;
```

Где x – номер программируемого вывода.

PORT_init_struct.GPIO_Mode – определяет режим работы вывода (выход или вход).

Записывается, например, так:

```
PORT_init_struct.GPIO_Mode = GPIO_Mode_OUT; //Вывод определяется как //выход.
```

PORT_init_struct.GPIO_Speed – определяет скорость работы вывода.

Например, так:

```
PORT_init_struct.GPIO_Speed = GPIO_Speed_2MHz;
```

Остальные 2 параметра записываются следующим образом:

```
PORT_init_struct.GPIO_OType = GPIO_OType_PP;  
PORT_init_struct.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

После заполнения структура должна выглядеть примерно так:

```
PORT_init_struct.GPIO_Pin = GPIO_Pin_12;  
PORT_init_struct.GPIO_Mode = GPIO_Mode_OUT;  
PORT_init_struct.GPIO_Speed = GPIO_Speed_2MHz;  
PORT_init_struct.GPIO_OType = GPIO_OType_PP;  
PORT_init_struct.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

Данную структуру надо записать в микроконтроллер с помощью строки:

```
GPIO_Init(GPIOx, &PORT_init_struct);
```

Где x – буква порта, вывод которого программируется.

Данная процедура повторяется для всех программируемых выводов.

Включение светодиода

Включение светодиода производится следующей строкой:

```
GPIO_SetBits(GPIOx, GPIO_Pin_y);
```

Где x – Буква порта. y – номер вывода, к которому подключен светодиод.

Выключение светодиода

Включение светодиода производится следующей строкой:

```
GPIO_ResetBits(GPIOx, GPIO_Pin_y);
```

Где x – Буква порта. y – номер вывода, к которому подключен светодиод.

Организация задержки

Задержка по времени работы микроконтроллера производится следующим образом:

В начале функции создаётся переменная i, с помощью которой и будет осуществляться задержка:

```
uint32_t i=0; //Примечание: переменные создаются в самом начале функции.
```

Затем, в нужном месте программы используется цикл for в качестве задержки программы на некоторое время.

```
for (i=0;i<3000000;i++) { }
```

Цикл вставляется в то место программы, где требуется остановить на время её выполнение.