Зажигание светодиода

Взглянем на плату. На плате, кроме всего прочего, имеется несколько светодиодов. Сейчас нам интересны те, которые расположены по кругу в нижней части платы (см. рисунок №1)

|  |
| --- |
|  |
| Рисунок №1 - Внешний вид платы и светодиоды.  [Источник – документация на плату STM32F3DISCOVERY] |

Рядом с каждым светодиодом написано его название. Выберем светодиод LD6. Первым делом нам необходимо найти куда подключен данный диод. Для этого необходимо открыть документацию на плату, найти в ней раздел про светодиоды (LEDs) и найти там выбранный светодиод. В документации указано, что выбранный светодиод (LD6) подключён к ножке PE15. Больше нам в этом разделе документация на плату не понадобится.

Чтобы зажечь светодиод необходимо сделать следующие шаги:

1. Подать питание на нужную нам группу ножек
2. Установить режим ножки, подключённой к светодиоду на «выход».
3. Выставить единицу на эту ножку

Итак, начнём с того, что все ножки объединены в группы, так называемые «банки». Они имеют имена A, B, C, D, E, F. Все эти банки вместе образуют так называемый GPIO (general-purpose input/output т.е. интерфейс ввода/вывода общего назначения). Узнать имя нужного нам банка можно из имени ножки. В нашем случае, ножка имеет имя PE15. Здесь “P” – значит pin, “E” – обозначает имя банка, к которому принадлежит данная ножка, “15” – обозначает номер ножки в банке. В каждом банке имеется 16 ножек с индексами от 0 до 15.

Выполним шаг первый и включим ножки банка E. Это действие выполняется путём редактирования специального регистра. Называется этот регистр RCC\_AHBENR (от enable register). Этот регистр изображен на рисунке №2

|  |
| --- |
|  |
| Рисунок №2 – Регистр RCC\_AHBENR. |

Находим в нём бит, включающий нужный нам банк. Называется от IOPEEN (input/output port E enable). В документации под изображением регистра есть более подробное описание всех его полей. Именно по этому описанию и нужно ориентироваться при поиске нужных полей регистра. Как видно, этот бит можно как считывать, так и записывать (rw). В описании к данному биту говорится, что для того, чтобы разблокировать GPIOE, нужно, чтобы в данном бите была единица. Чтобы заблокировать – нуль. Код для включения будет выглядеть так:

RCC->AHBENR |= (1 << 21); //Подали питание на GPIOE

Не будем подробно останавливаться на том как это работает (код). В случае непонимания обращаться к части «Битовые операции».

Теперь нам нужно настроить режим ножки 15. Режим настраивается через регистр GPIOE\_MODER (mode register). Этот регистр приведён на рисунке №3

|  |
| --- |
|  |
| Рисунок №3 – Регистр GPIOE\_MODER. |

Видно, что для каждой ножки для описания режима выделено два бита. Все биты в данном регистре можно считывать и записывать. В описании к регистру сказано, что режим задаётся по следующему правилу:

00 – режим «входа»

01 – режим «выхода»

10 – режим «альтернативной функции»

11 – аналоговый режим.

Нам нужно настроить ножку на режим «выхода» так как мы сами будем задавать значение, которое должно быть на ножке. Режим «входа» нужен, когда мы рассчитываем считывать информацию с ножки. Режим «альтернативной функции» понадобится нам далее и мы не будем его здесь рассматривать. Итак, нам необходимо, чтобы в бите 30 находилась единица, а в бите 31 – нуль (тогда поле MODER15 будет иметь значение 01). Для этого будем использовать следующий код:

GPIOE->MODER &= ~(1 << 31); //Устанавливаем нуль в бит 31

GPIOE->MODER |= (1 << 30); // Устанавливаем единицу в бит 30

Теперь осталось только подать на эту ножку единицу и светодиод должен загореться. За значения, которые мы устанавливаем на ножке отвечает регистр GPIOE\_ODR (output data register). Его поля можно увидеть на рисунке №4.

|  |
| --- |
|  |
| Рисунок №4 – Регистр GPIOE\_ODR. |

Чтобы на ножке 15 была выставлена единица, нужно поместить в бит ODR15 единицу. Обращаем внимание, что бит этот разрешено записывать и считывать. Код будет выглядеть следующим образом:

GPIOE->ODR |= (1 << 15); //Зажигаем светодиод.

Итак, программа целиком будет выглядеть примерно таким образом:

#include "stm32f3xx.h"

int main(void)

{

RCC->AHBENR |= (1 << 21); //Разрешаем работу GPIOE

GPIOE->MODER &= ~(1 << 31); //Настраиваем режим ножки 15

GPIOE->MODER |= (1 << 30);

GPIOE->ODR |= (1 << 15); //Зажигаем светодиод.

while(1) //зацикливаемся здесь, чтобы процессор не пытался больше ничего делать

{

}

return 0;

}