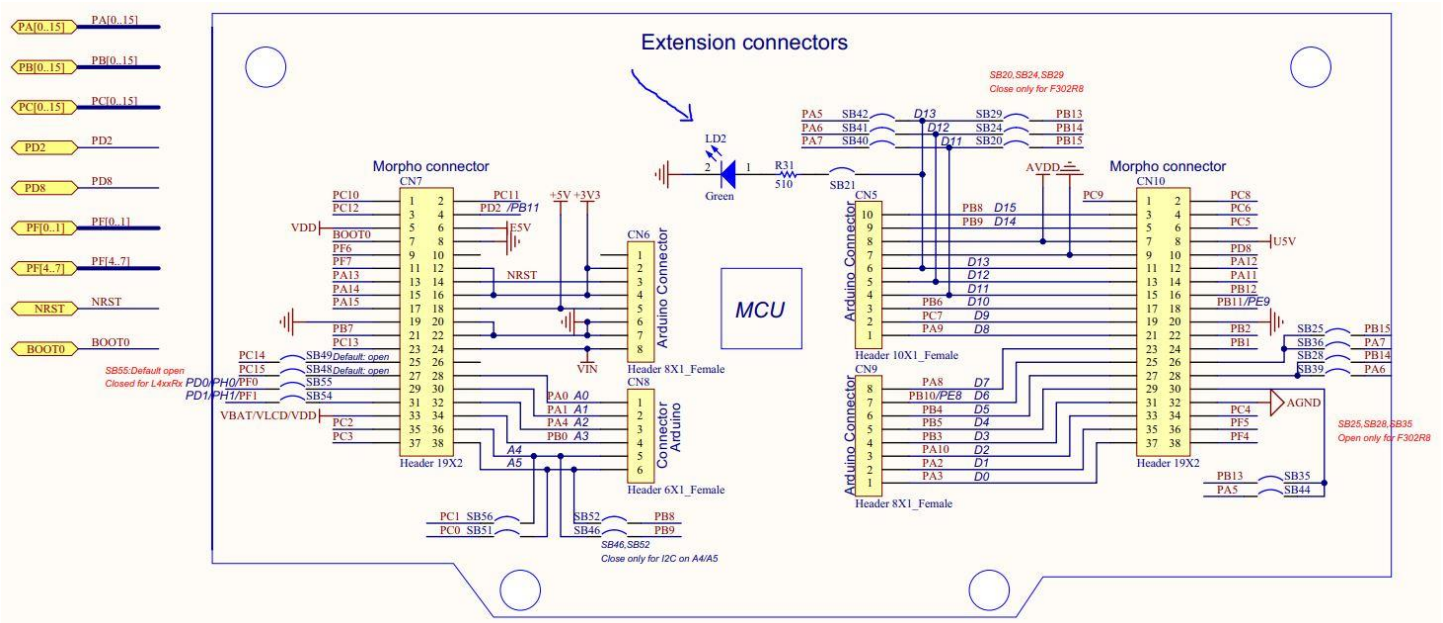
Code for STM32f411 Nucleo-64 to make Green LED light on (include Assembly)

Feb 22, 2020 Dung-Yi Chao

1. From the STM32 Nucleo-64 boards user Manual (link), you should first locate the port and pin which connect to the green LED (LD2). We can tell from the following image (from the manual) that LD2 connects to PA5 (GPIOA port, pin5)



2. We now first enable the clock for GPIOA according to the following image. Code: **RCC->AHB1ENR |= 1U;**

RM0383 **Reset and clock control (RCC) for STM32F411xC/E**

6.3.9 RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

Address offset: 0x30

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DMA2EN	DMA1EN	Reserved				
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved				GPIOHEN	Reserved		GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
			rw					rw			rw	rw	rw	rw	rw

3. Set the mode for GPIOA Pin5. From the following image, you should set MODER5[1:0] to 01 as output mode. Code: **GPIOA->MODER |= (1U<<10)**;

8.4.1 GPIO port mode register (GPIOx_MODER) (x = A..E and H)

Address offset: 0x00

Reset values:

- 0xA800 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

4. Now, you are able to make the LED light up. LD2 is pin 5, so you need to make the 5th bit to 1. Code: **GPIOA->ODR |= (1U<<5);**

8.4.6 GPIO port output data register (GPIOx_ODR) (x = A..E and H)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

5. The overall code will be the following:

```
#include "stm32f4xx.h"
#define GREEN (1U<<5)
#define GPIOA_CLOCK (1U)

int main()
{
        RCC->AHB1ENR |= GPIOA_CLOCK;
        GPIOA->MODER |= (1U<<10);

        while(1)
        {
                GPIOA->ODR |= GREEN;
        }
}
```

6. Assembly code part

A. First, you need to locate the address of **RCC_AHB1ENR** and **GPIOA**. From the following table, you can find the base address of RCC is 0x40023800 and GPIOA is 0x40020000.

2.3 Memory map

See the datasheet corresponding to your device for a comprehensive diagram of the memory map. *Table 1* gives the boundary addresses of the peripherals available in STM32F411xC/E device.

Table 1. STM32F411xC/E register boundary addresses

Boundary address	Peripheral	Bus	Register map
0x5000 0000 - 0x5003 FFFF	USB OTG FS	AHB2	*Section 22.16.6: OTG_FS register map on page 754*
0x4002 6400 - 0x4002 67FF	DMA2		*Section 9.5.11: DMA register map on page 197*
0x4002 6000 - 0x4002 63FF	DMA1		
0x4002 3C00 - 0x4002 3FFF	Flash interface register		*Section 3.8: Flash interface registers on page 59*
0x4002 3800 - 0x4002 3BFF	RCC		*Section 6.3.22: RCC register map on page 136*
0x4002 3000 - 0x4002 33FF	CRC	AHB1	*Section 4.4.4: CRC register map on page 69*
0x4002 1C00 - 0x4002 1FFF	GPIOH		*Section 8.4.11: GPIO register map on page 163*
0x4002 1000 - 0x4002 13FF	GPIOE		
0x4002 0C00 - 0x4002 0FFF	GPIOD		
0x4002 0800 - 0x4002 0BFF	GPIOC		
0x4002 0400 - 0x4002 07FF	GPIOB		
0x4002 0000 - 0x4002 03FF	GPIOA		

Thus, **RCC_AHB1ENR**'s address can be calculated by looking at the following image (ie. 0x40023800 + 0x30 = 0x40023830). GPIOA_MODER and GPIOA_ODR can also be calculated in this manner.

6.3.9 RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

Address offset: 0x30

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DMA2EN	DMA1EN	Reserved				
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved				GPIOHEN	Reserved		GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
			rw					rw			rw	rw	rw	rw	rw

Bit 0 **GPIOAEN:** IO port A clock enable

Set and cleared by software.

0: IO port A clock disabled

1: IO port A clock enabled

B. Two pieces of code **initports.s** (subroutine) and **main.s** are in the following.

main.s

```
GPIOA_ODR       EQU 0x40020014

                AREA    ARMex, CODE, READONLY
                ENTRY
__main  PROC
                 EXPORT  __main
                 IMPORT initports
                bl initports

loop
                b light_on


light_on
                ldr r0, =GPIOA_ODR
                MOV r1, #0x20
                str r1, [R0]
                b loop


                ENDP
                END
```

initports.s

```
GPIOA_MODER     EQU   0x40020000
RCC_AHB1ENR     EQU   0x40023830

                        AREA    ARMex, CODE, READONLY
initports    PROC
                  EXPORT  initports

                push {lr}

                LDR r5, = RCC_AHB1ENR
                 LDR r6, [r5]
                ORR r6, #0x1
                 STR r6, [r5]
                LDR r5, = GPIOA_MODER
                 LDR r6, [r5]
                ORR r6, #0x400
                 STR r6, [r5]

                 pop {lr} ;restore link register
                bx lr   ;return from subroutine

                ENDP
                END
```