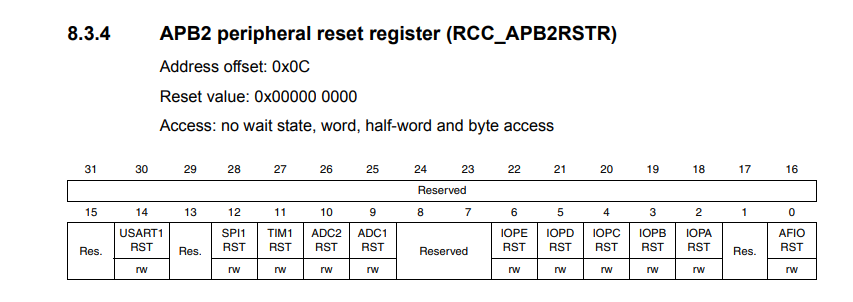
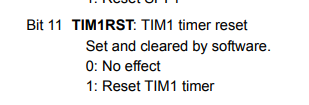
1. The peripherals’ clocks are disabled by default. If you need to use a peripheral you need to enable its clock manually. Why are the peripherals' clock disabled by default?

To save power, turning off the clock when its not directly used saves energy making these microcontrollers more energy efficient

1. RCC\_APB2RSTR (Advanced Peripheral Bus 2 Reset Register) is used to disable the clock to GPIO ports. Write the code to disable the clock to the LEDs port without changing any other bit in that register.

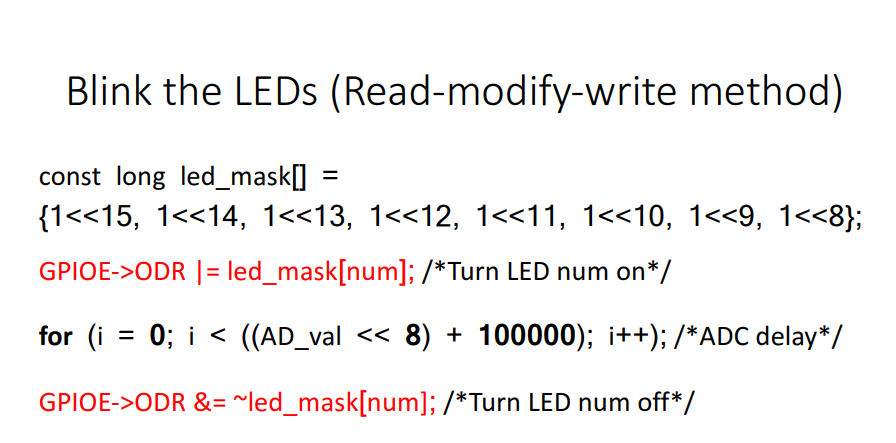




So I have to change TIM1 only so: RCC->APB2RSTR 1<<11

Does this actually diable the timer or is it something else? Page 140

1. Write the proper code (in C language) to turn on LED 2 and LED 5 using the Readmodify-write method (Clock + configurations) Explain



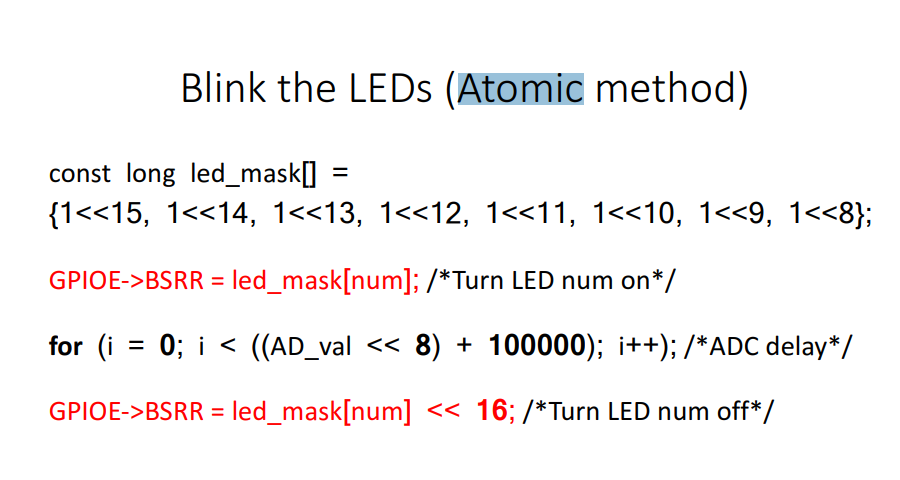
Assumptions: this led\_mask[] works for all LEDs and that the num referenced in the 3ird line reference the numbered LED based on this

Using the mask above

GPIOE->ODR |= led\_mask[2]; //turns LED2 on

GPIOE->ODR |= led\_mask[5]; //turns LED5 on

As for enabling the clock I have no idea which ports to use or how to find out which ports to use  
something to do with RCCAPB2ENR page 145

1. Write the proper code (in C language) to turn on LED 1 and LED 6 using the Atomic method (Clock + configurations) Explain.

Same as 3

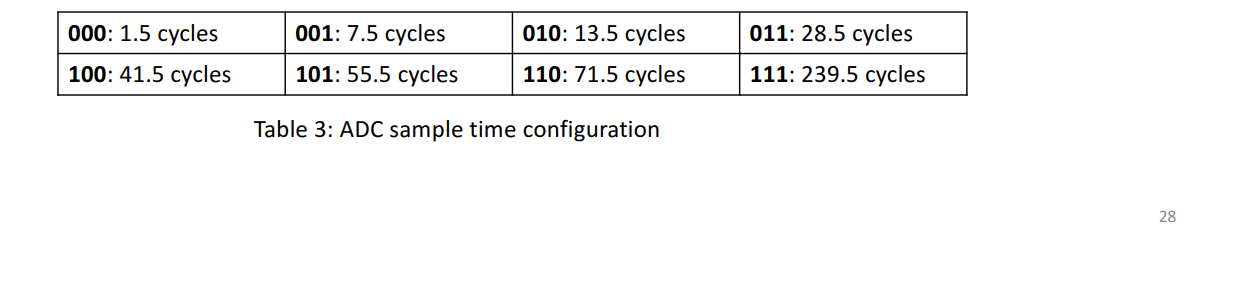
5. An ADC clock is running at 20 MHz, for a sample time of 4 uS, configure the SMPR register accordingly.

T=1/f = 1/20MHz = 0.05us

Total time delayed – 4us

4us/0.05us = 80 cycles

Use table 3



Using 111 over 110 because I don’t want the delay to be too short?

ADC! -> SMPR! = 7 << 12;

Is this it?