STM32 Timers

G V V Sharma*

Contents

***Abstract*—This manual shows how to program timers in arm using STM32F103C8T6.**

1 Components

Component	Value	Quantity
Breadboard		1
Resistor	220 Ω	1
STM32F103C8T6		1
Seven Segment Display	Common Anode	1
Jumper Wires		20

TABLE 1.0

Problem 1.1. List all available clocks in the STM32F103C8T6 blue pill.

Solution: See Table 1.1.

Clock	Location	Type	Frequency
HSI	Internal	RC	8Mhz
LSI	Internal	RC	32.768 kHz
HSE	Internal	Crystal	8Mhz

TABLE 1.1

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

2 Systick timer

The Systick timer is usually available on ARM chips and is different from the timers provided by the manufacturer.

Problem 2.1. Execute the program in

```
https://github.com/gadepall/
    STM32F103C8T6/blob/master/
    examples/blink_systick.c
```

Problem 2.2. The default clock is the HSI 8MHz RC. Find the number of clock cycles required for a 1 s delay.

Solution: The time period is

$$T = \frac{1}{8}\mu s = 1 \text{ cycle} \tag{2.2.1}$$

Thus, the number of cycles required for 1 s delay is

$$1 \text{ second} = 8000000 \text{ cycles} \tag{2.2.2}$$

Problem 2.3. List the SysTick registers.

Solution: See Table 2.3.

Register	Command	Purpose
SysTick Control and Status	SysTick->CTRL	Timer control
SysTick Reload Value	SysTick->LOAD	Timer Count
SysTick Current Value	SysTick->VAL	Timer Initialize
SysTick Calibration Value		

TABLE 2.3

Problem 2.4. What do the following instructions do?

```
SysTick->LOAD = 4000000;
SysTick->VAL = 0;
```

Solution: See Table 2.3 for details. These two instructions ask the SysTick timer to count down from 4000000 to 0.

Problem 2.5. Explain the following instruction.

```
while(!(SysTick->CTRL & 0x00010000
    ));
```

Solution: Fig. 2.5 shows the SysTick CTRL register. 0x00010000 is used in the above command to mask all the bits except for bit 16, which is the COUNTFLAG. The **while** loop will stop once COUNTFLAG = 0. The while loop is used for the delay. will stop once

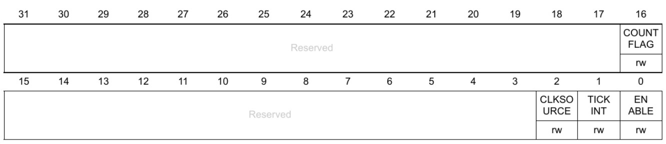

Fig. 2.5

Problem 2.6. What does the following instruciton do?

```
SysTick->CTRL = 0x00000005;          //
    8MHz clock
```

Solution: From Fig. 2.5, ENABLE = 1 enables the counter (for delay) and CLKSOURCE = 1 enables the 8 MHz internal RC clock.

Problem 2.7. Obtain a 1 MHz clock.

Solution: CLKSOURCE = 1 results in the $\frac{\text{Processor Clock}}{8}$ = 1 MHz clock.

```
SysTick->CTRL = 0x00000001;          //
    1MHz clock
```

Problem 2.8. Obtain a delay of 1 second using the 1 MHz clock.

3 Hardware

The STM32F103C8T6 micro-controller in Fig. 3.1 has two ground pins, few analog input pins and few digital pins that can be used for both input as well as output. It has one Vcc (3.3V) pin that can generate 3.3V. In the following exercises, only the GND, 3.3V and digital pins will be used.

Problem 3.1. Make the pin connections in Table 3.1 using Figs. **??** and 3.1.

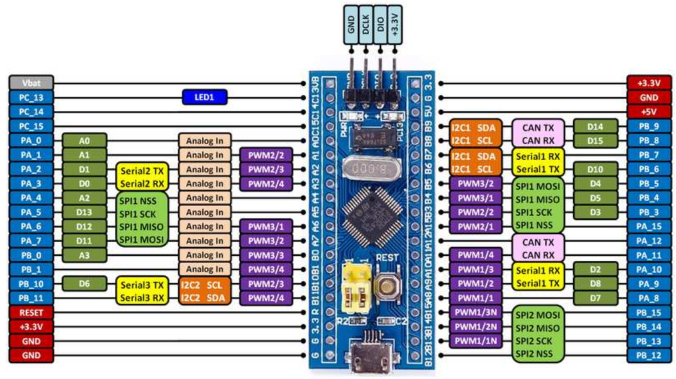

Fig. 3.1

STM32	PB9	PB8	PB7	PB6	PB5	PB4	PB3	3.3V
Display	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**COM**
0	0	0	0	0	0	0	1	
1	1	0	0	1	1	1	1	
2	0	0	1	0	0	1	0	
3	0	0	0	0	1	1	0	
4	1	0	0	1	1	0	0	
5	0	1	0	0	1	0	0	
6	0	1	0	0	0	0	0	
7	0	0	0	1	1	1	1	
8	0	0	0	0	0	0	0	
9	0	0	0	0	1	0	0	

TABLE 3.1

4 Software

Problem 4.1. Execute

```
https://github.com/gadepall/
    STM32F103C8T6/blob/master/
    examples/sevenseg_example.c
```

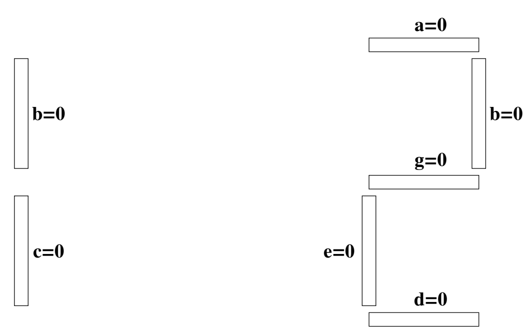

Fig. 4.1

Problem 4.2. Explain the process of generating the number 0 using the following instruction.

```
GPIOB->ODR = 0xFC08;
```

Solution: ODR is the Output Data Register, which is used to write outputs to the GPIO pins. The 16 bit number 0xFC08 on the RHS represents the pin configuration for the pins of port B of STM32F103C8T6, which are numbered PB15-PB0 in that order. See Table 3.1,

Problem 4.3. Repeat the above exercise to generate the numbers 1-9 on the display.

Problem 4.4. The previous instructions set the bits in the unused ports PB15-PB10 and PB2-PB0. This may be undesirable in some cases. Generate 0 by not disturbing the unused pins.

Solution: The following instructions help accomplish this. The first instruction resets PB4-PB9. The second instruction sets the PB3 pin. The other pins are undisturbed.

```
GPIOB->BRR = (1<<4)|(1<<5)|(1<<6)
    |(1<<7)|(1<<8)|(1<<9);  // (Led
    ON)
GPIOB->BSRR = (1<<3);  // (Led OFF)
```

Problem 4.5. Write a program to take a 4-bit BCD as input from hardware (GND or VDD) and show the next number on the seven segment display.

Solution: The following program takes 4 bits as input from pins PB12-PB15 and displays the output on a seven segment display. The next number can be displayed by slightly modifying the code.

```
https://github.com/gadepall/
    STM32F103C8T6/blob/master/
    examples/bin2dec_example.c
```