Seven Segment Display control through STM32 GPIO

Alok Ranjan Kesari and G V V Sharma*

Contents

***Abstract*—The objective of this manual is to introduce beginners to arm embedded programming by powering a seven segment display.**

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

2 Hardware

The seven segment display in Fig. 2.1 has eight pins, a, b, c, d, e, f, g and *dot* that take an active LOW input, i.e. the LED will glow only if the input is connected to ground. Each of these pins is connected to an LED segment. The *dot* pin is reserved for the · LED.

Problem 2.1. Connect one end of the 1K resistor to the COM pin of the display and the other end to an extreme pin of the breadboard.

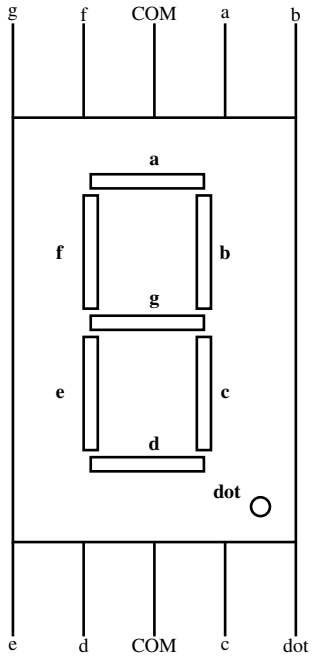

Fig. 2.1

The STM32F103C8T6 micro-controller in Fig. 2.2 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 2.2. Make the pin connections in Table 2.2 using Figs. 2.1 and 2.2.

3 Software

Problem 3.1. Execute

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

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

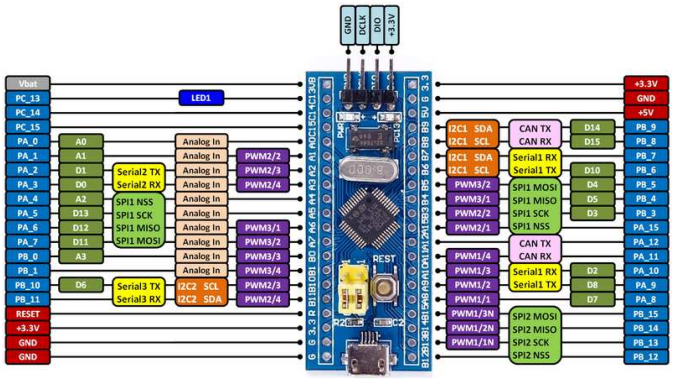

Fig. 2.2

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 2.2

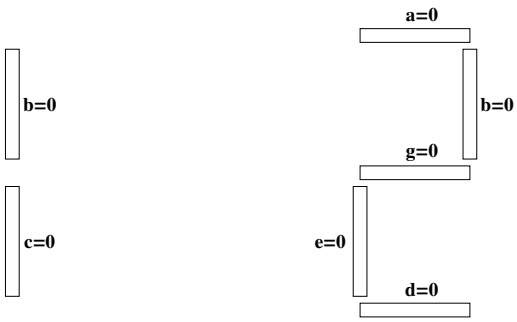

Fig. 3.1

Problem 3.2. 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 2.2,

Problem 3.3. Repeat the above exercise to generate the numbers 1-9 on the display.

Problem 3.4. 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 3.5. 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/sevenseg/
    bin2dec_example.c
```