

Alok Ranjan Kesari and G V V Sharma*

CONTENTS

1	Components	1
2	Connecting the Display	1
3	Display Control with STM32	2

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 CONNECTING THE DISPLAY

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 \cdot 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.

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

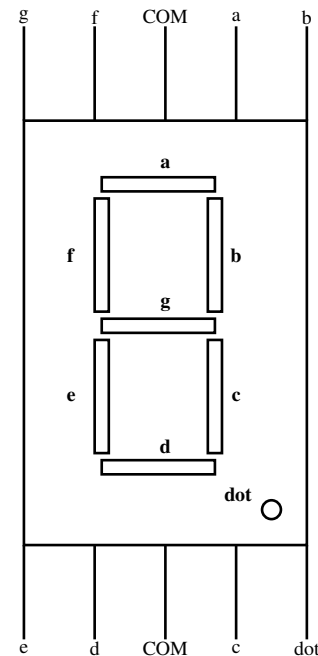


Fig. 2.1

STM32	PB15	PB14	PB13	PB12	PB11	PB10	PB9
Display	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1

TABLE 2.2

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.

Problem 2.3. Connect the STM32 to the Raspberry

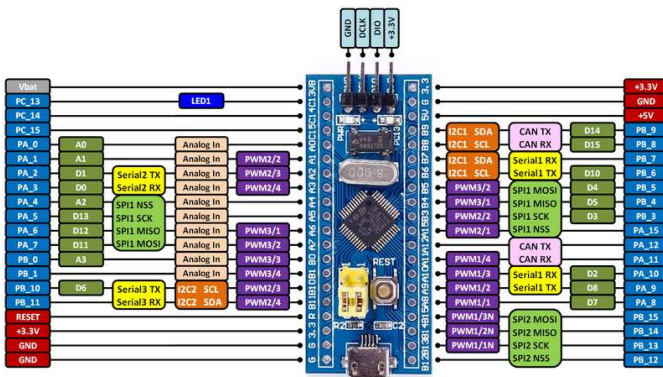


Fig. 2.2

Pi according to the instructions in

```
https://github.com/gadepall/EE4013/
blob/master/setup/rpi/
gvv_stm32_setup.pdf
```

3 DISPLAY CONTROL WITH STM32

Problem 3.1. Follow the instructions in

```
https://github.com/gadepall/EE4013/
blob/master/setup/tinker/
gvv_stm32_tinker_setup.pdf
```

to clone the repository

```
https://github.com/gadepall/
STM32F103C8T6
```

Fig. 3.2 explains how to get decimal digits using the seven segment display.

Problem 3.2. In the STM32F103C8T6 directory,

```
cp sevenseg.c main.c
```

Generate the number 0 by executing **main.c** and flashing to the STM32.

Problem 3.3. Explain the process of generating the number 0 in the above code.

Solution: The relevant line in **sevenseg.c** is

```
GPIOB->ODR = 0x03FF; // 0
```

The hexadecimal number on the RHS represents the pin configuration for the pins of port B of STM32F103C8T6, which are numbered PB15-PB0 in that order. For generating the number 0, according

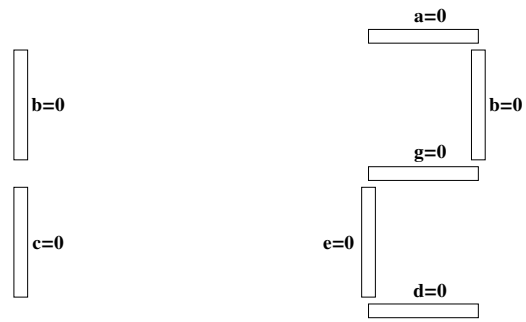


Fig. 3.2

to Table 2.2, Pins PB15-PB8 result in 0x03, since pin PB8 is kept high. Also, pins PB7-PB0 are also kept high, resulting in FF.

Problem 3.4. Repeat the above exercise to generate the numbers 1-9 on the display.

Problem 3.5. Introduce a delay to generate a decade counter.