

CONTENTS

1	Display Control through Hardware	1
1.1	Components	1
1.2	Software Setup	1
1.3	Powering the Display	1
1.4	Controlling the Display . . .	2
2	Display Control through Software	2
2.1	Driving the Segments	2

Abstract—The objective of the manual is to introduce the RaspberryPi to beginners by programming the GPIO pins to build a decade counter .

1 DISPLAY CONTROL THROUGH HARDWARE

1.1 Components

The components required for this manual are listed in Table 1.0

Component	Value	Quantity
Breadboard		1
Resistor	$\geq 220\Omega$	1
Pi	Model B, Rev 3	1
Seven Segment Display	Common Anode	1
Jumper Wires	Female-Male	20

TABLE 1.0

*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.

1.2 Software Setup

The following commands will install the Wiring Pi module

```
sudo apt-get install git-core
sudo apt-get update
sudo apt-get upgrade
cd
git clone git://git.drogon.net/
wiringPi
cd ~/wiringPi
./build
```

1.3 Powering the Display

The breadboard can be divided into 5 segments. In each of the green segments, the pins are internally connected so as to have the same voltage. Similarly, in the central segments, the pins in each column are internally connected in the same fashion as the blue columns.

Problem 1.1. Plug the display to the breadboard in Fig. 1.1

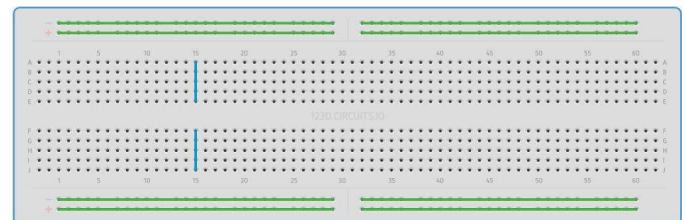


Fig. 1.1

The seven segment display in Fig. 1.2 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 1.2. Connect one end of the resistor to the COM pin of the display and the other end to an extreme pin of the breadboard.

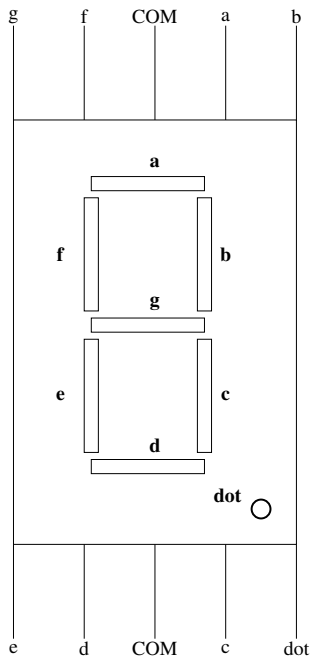


Fig. 1.2

The Raspberry Pi 3 has 40 pins (see Figs. 2.1.1 and 2.1.2), which include power pins that can generate 5V and 3.3V, GND pins, PWM pins, pins for wired communication and some free pins for digital I/O. In the following exercises, only the GND, 5V and digital I/O pins will be used.

Problem 1.3. Connect pin 2 (5V) of the Pi to an extreme pin that is in the same segment as the resistor pin.

Problem 1.4. Connect pin 6 (GND) of the Pi to the opposite extreme pin of the breadboard

Problem 1.5. Connect the *dot* pin of the display to a pin in the same segment as the GND pin. What do you observe?

1.4 Controlling the Display

Fig. 1.6 explains how to get decimal digits using the seven segment display.

Problem 1.6. Generate the number 1 on the display by connecting only the pins *b* and *c* to GND.

Problem 1.7. Repeat the above exercise to generate the number 2 on the display.

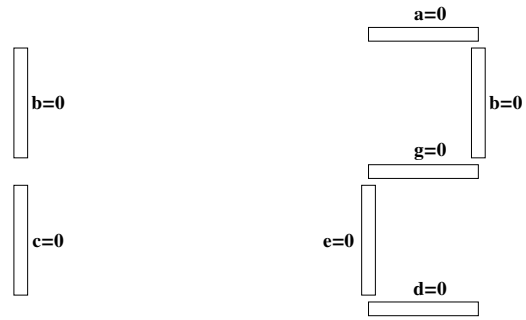


Fig. 1.6

Problem 1.8. Table 1.8 summarizes the process of generating the decimal digits. 0 means connecting to ground and 1 means not connecting. Complete Table 1.8 for all numbers between 0-9.

a	b	c	d	e	f	g	decimal
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2

TABLE 1.8

Problem 1.9. Now generate all numbers between 0-9 on the display using the above table.

2 DISPLAY CONTROL THROUGH SOFTWARE

2.1 Driving the Segments

Problem 2.1.



Fig. 2.1.1: GPIO pin snapshot on Pi.

Problem 2.2. Connect the a-g pins of the display to the GPIO pins 0-6 of the Pi shown in 2.1.1 and 2.1.2.

Problem 2.3. Type the following C code and excute. What do you observe?

Solution:

Raspberry Pi 3 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5		6	Ground
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART) 15
	Ground	9		10	GPIO 16 RxD (UART) 16
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0 1
2	GPIO 2	13		14	Ground
3	GPIO 3	15		16	GPIO 4 4
	3.3 VDC Power	17		18	GPIO 5 5
12	GPIO 12 MOSI (SPI)	19		20	Ground
13	GPIO 13 MISO (SPI)	21		22	GPIO 6 6
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI) 10
	Ground	25		26	GPIO 11 CE1 (SPI) 11
30	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM) 31
21	GPIO 21 GPCLK1	29		30	Ground
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0 26
23	GPIO 23 PWM1	33		34	Ground
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27 27
25	GPIO 25	37		38	GPIO 28 PCM_DIN 28
	Ground	39		40	GPIO 29 PCM_DOUT 29

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Fig. 2.1.2: GPIO Wiring Pi pin configuration.

```

}

int main (void)
{
    wiringPiSetup () ;
    pinMode (0, OUTPUT) ;
    pinMode (1, OUTPUT) ;
    pinMode (2, OUTPUT) ;
    pinMode (3, OUTPUT) ;
    pinMode (4, OUTPUT) ;
    pinMode (5, OUTPUT) ;
    pinMode (6, OUTPUT) ;

    for (;;)
    {
        sevensseg (1,0,0,1,1,1,1);
    }
}
//Command for raspberry pi
//gcc -Wall -o test_seven_seg_disp
//.c -lwiringPi
//followed by
// sudo ./test

```

Problem 2.4. Now generate the numbers 0-9 by modifying the above program.

Problem 2.5. Suitably modify the above program to obtain a decade counter.

```

#include <wiringPi.h>

void sevensseg(int a,int b,int c,
    int d,int e,int f,int g)
{
    digitalWrite(0, a);
    digitalWrite(1, b);
    digitalWrite(2, c);
    digitalWrite(3, d);
    digitalWrite(4, e);
    digitalWrite(5, f);
    digitalWrite(6, g);
}

```