Digital Design through Pi

G V V Sharma*

Contents

1 Display Control through Hardware

1.1 Powering the Display

Problem 1.1. Plug the display to the breadboard shown in Fig. 1.1. The breadboard can be divided into 5 segments. In each of the green segements, 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.

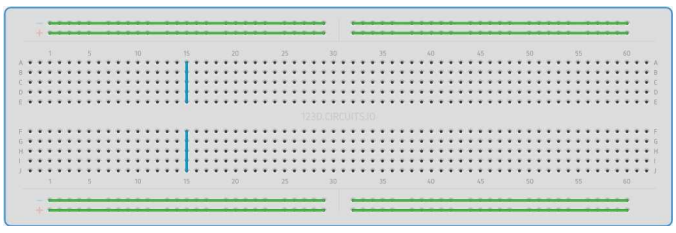

Fig. 1.1: Breadboard

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

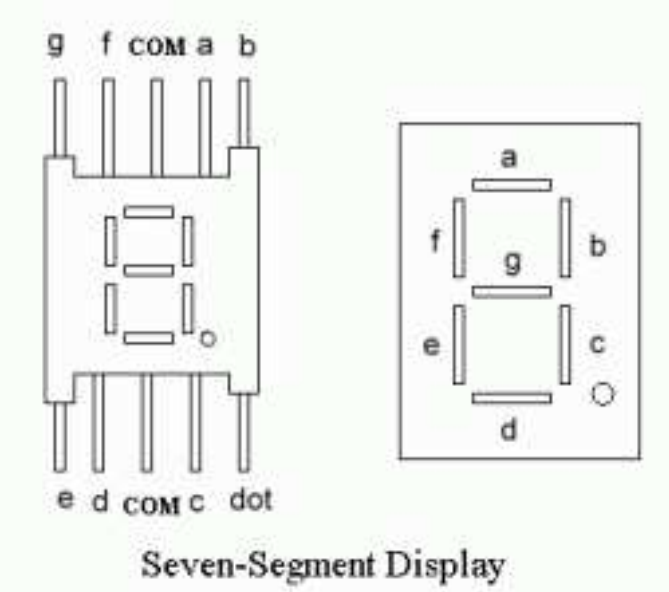

Fig. 1.2: Seven Segment Display.

Problem 1.2. Connect one end of the 220 Ω resistor to the COM pin of the display in Fig. 1.2 and the other end to an extreme pin of the breadboard. The seven segment display 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 1.3. Connect the $3.3V$ pin of the Pi shown in Figs. 1.3.1 and 1.3.2 to an extreme pin that is in the same segment as the 220 Ω resistor pin.

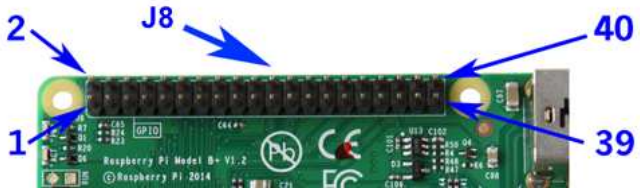

Fig. 1.3.1: GPIO pin snapshot on Pi [1].

TABLE 1.6: Seven segment display to decimal.

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

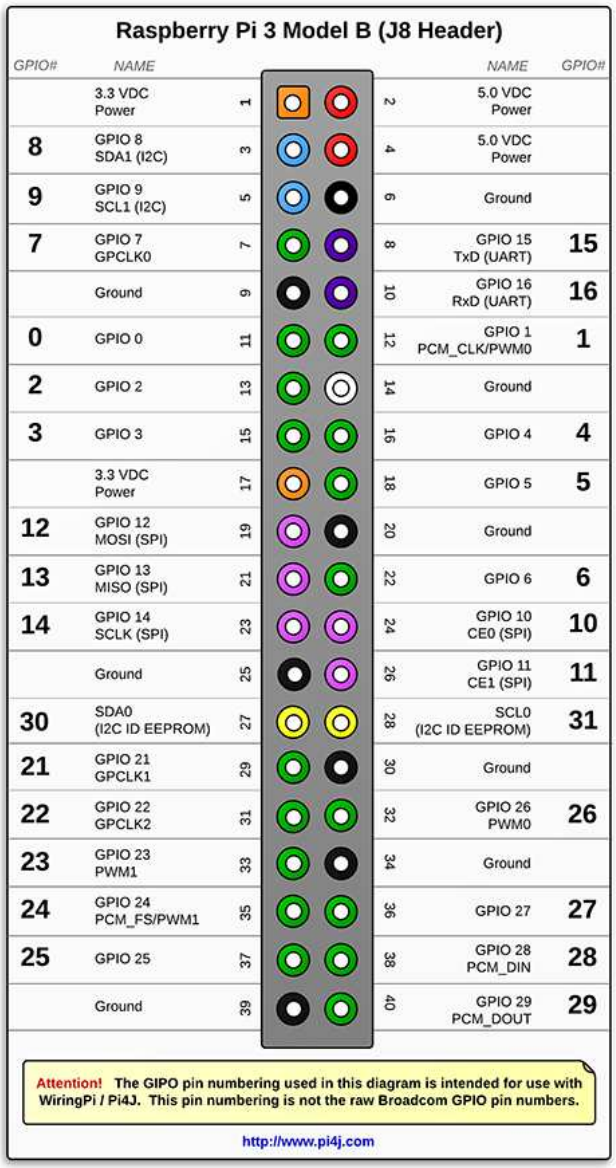

Fig. 1.3.2: GPIO Wiring Pi [2] pin configuration.

Problem 1.4. Connect the GND pin 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?

1.2 Controlling the Display

Problem 1.6. Generate the number 1 on the display by connecting the pins $a - g$ to GND according to Table 1.6.

Problem 1.7. Complete Table 1.6 for all numbers between 0-9.

Problem 1.8. Now generate the numbers from 0-9 on the display using Table 1.6.

Problem 1.9. Connect the 7447 IC decoder $\bar{a} - \bar{g}$ pins in Fig. 1.9 to the $a - g$ pins of the display respectively.

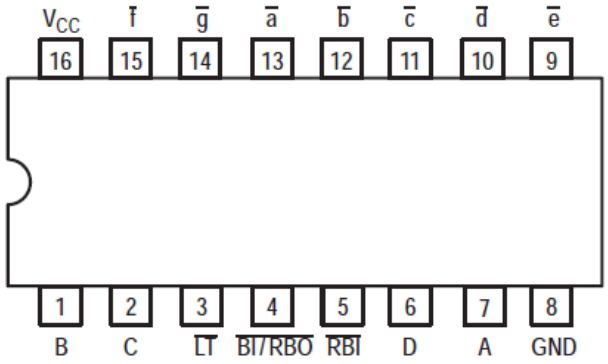

Fig. 1.9: 7447 to seven segment display decoder.

Problem 1.10. Connect the V_{cc} and GND pins of the decoder in to the supply and GND pins of the breadboard.

Problem 1.11. Connect the A,B,C,D pins in Fig. 1.9 to pins in the GND extreme segment of the breadboard. What do you observe?

Problem 1.12. Now remove the D pin from the breadboard and observe the display output.

Problem 1.13. Generate a table with A,B,C,D inputs and the equivalent decimal number output.

The 7447 IC helps in displaying decimal numbers on the seven segment display. The $\bar{a} - \bar{f}$, pins of the 7447 IC are connected to the $a - f$ pins of the display. V_cc should be connected to a 5V power source. The input pins of the decoder are A,B,C and D, with A being the lowest significant bit (LSB) and D being the most significant bit (MSB). For example, the number 5 is visible on the display when the A,B,C and D inputs are the following.

D	C	B	A	Decimal
0	1	0	1	5

2 Software Setup

Problem 2.1. Run the following commands to install WiringPi.
sudo apt-get install git-core
sudo apt-get update
sudo apt-get upgrade

Problem 2.2. Download geany using
sudo apt-get install geany

3 Display Control through Software

Problem 3.1. Connect the A-D pins of the 7447 IC in Fig. 1.9 to the GPIO pins 0-3 of the Pi shown in Figs. 1.3.1 and 1.3.2.

Problem 3.2. Execute the following code using the following commands:
gcc -Wall -o test bcd_seven.c -lwiringPi
sudo ./test
What do you observe?

```
#include <wiringPi.h>
int main (void)
{
  wiringPiSetup () ;
  pinMode (0, OUTPUT) ;
  pinMode (1, OUTPUT) ;
  pinMode (2, OUTPUT) ;
  pinMode (3, OUTPUT) ;
  for (;;)
  {
   digitalWrite (0, 1);
   digitalWrite (1, 0) ;
   digitalWrite (2, 1) ;
   digitalWrite (3, 0) ;
  }
  return 0 ;
}
//Command for raspberry pi
//gcc -Wall -o test bcd_seven.c -
   lwiringPi
//followed by
// sudo ./test
```

Problem 3.3. Now generate the numbers 0-9 by modifying the above program.

Problem 3.4. After the following line in the previous code,

```
int main (void) {
```

you can define integer variables as

```
int A = 0;
```

where the variable A is defined to be an integer and given the value 0. Define variables A,B,C,D as 0 or 1 and use the digitalWrite() command as in the earlier code to generate the numbers 0-9.

4 Combinational Logic

4.1 Counting Decoder

Table 4.1 represents the system that increments the numbers 0-8 by 1 and resets the number 9 to 0. Such a table is known as a truth table, where W, X, Y, Z are the inputs and A, B, C, D are the outputs. Note that $D = 1$ for the inputs 0111 and 1000. Using *boolean* logic,

$$D = WXYZ' + W'X'Y'Z \tag{1}$$

Note that 0111 results in the expression $WXYZ'$ and 1000 yields $W'X'Y'Z$.

Problem 4.1. Write the boolean logic functions for A, B, C in terms of W, X, Y, Z.

The && operand is used for the boolean AND (multiplication) operation, the || operand is used for the OR (addition) operation and the ! operand is used for the NOT ($'$) operation in Arduino code. For example, the expression for (1) in Arudino is

```
D = (W&&X&&Y&&!Z)||(!W&&!X&&!Y&&Z);
```

Z	Y	X	W	D	C	B	A
0	0	0	0	**0**	**0**	**0**	**1**
0	0	0	1	**0**	**0**	**1**	**0**
0	0	1	0	**0**	**0**	**1**	**1**
0	0	1	1	**0**	**1**	**0**	**0**
0	1	0	0	**0**	**1**	**0**	**1**
0	1	0	1	**0**	**1**	**1**	**0**
0	1	1	0	**0**	**1**	**1**	**1**
0	1	1	1	**1**	**0**	**0**	**0**
1	0	0	0	**1**	**0**	**0**	**1**
1	0	0	1	**0**	**0**	**0**	**0**

TABLE 4.1

Problem 4.2. Write the code for the outputs A, B, C and verify if your logic is correct by observing the output on the seven segment display.

4.2 Display Decoder

Problem 4.3. Now write the truth table for the seven segment display decoder (IC 7447). The inputs will be A, B, C, D and the outputs will be a, b, c, d, e, f, g.

Problem 4.4. Obtain the logic functions for outputs a, b, c, d, e, f, g in terms of the inputs A, B, C, D.

Problem 4.5. Disconnect the Pi from IC 7447 and connect the pins 0-6 in Fig. 1.3.2 in the Pi directly to the seven segment display.

Problem 4.6. Write a new program to implement the logic in Problem 4.4 and observe the output in the display. You have designed the logic for IC 7447!

Problem 4.7. Now include your counting decoder program in the display decoder program and see if the display shows the consecutive number.

4.3 Software Counter

Problem 4.8. Connect pin 10 to the dot pin of the display and execute the following program. What do you observe?

```
#include <wiringPi.h>
int main (void)
{
  wiringPiSetup () ;
  pinMode (10, OUTPUT) ;
  for (;;)
  {
    digitalWrite (10, HIGH) ;
      delay (500) ;
    digitalWrite (10, LOW) ;
      delay (500) ;

  }
  return 0 ;
}
//Command for raspberry pi
//gcc -Wall -o blink blink.c -
   lwiringPi
//followed by
// sudo ./blink
```

Problem 4.9. A decade counter counts the numbers from 0-9 and then resets to 0. Suitable modify the above program to obtain a decade counter.

5 Sequential Logic

5.1 Decade Counter through Flip Flops

The 7474 IC has two D flip flops. The D pins denote the input and the Q pins denote the output. CLK denotes the clock input.

Problem 5.1. Connect the 0-3 pins of the Pi to the Q pins of the two 7474 ICs. Use the 0-3 pins as Pi input.

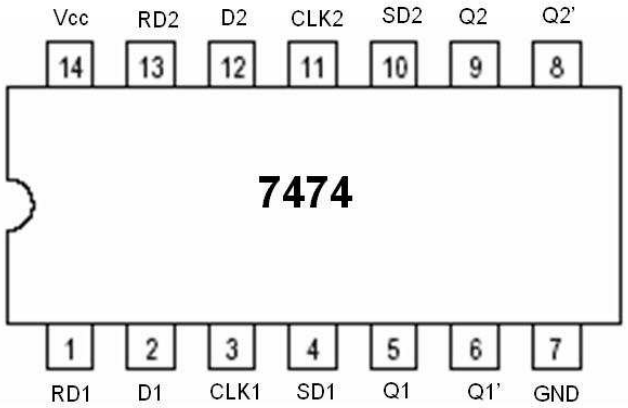

Fig. 5.1: Flip flop

Problem 5.2. Connect the Q pins to IC 7447 Decoder as input pins. Connect the 7447 IC to the seven segment display.

Problem 5.3. Connect the 11-14 pins of the Pi to the D input pins of two 7474 ICs. Use the 11-14 pins as Pi output.

Problem 5.4. Connect pin 10 of the Pi to the CLK inputs of both the 7474 ICs.

Problem 5.5. Using the logic for the counting decoder in the Pi software, implement the decade counter.

References

[1] [Online]. Available: http://pi4j.com/pins/model-2b-rev1.html
[2] [Online]. Available: http://wiringpi.com/