Digital Design through Arduino

G V V Sharma*

Contents

***Abstract*—This manual covers the entire breadth of digital design by building a decade counter using an arduino. In the process, boolean logic, combinational logic and sequential logic are covered.**

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
Arduino	Uno	1
Seven Segment Display	Common Anode	1
Decoder	7447	1
Flip Flop	7474	2
Jumper Wires		20

TABLE 1.0

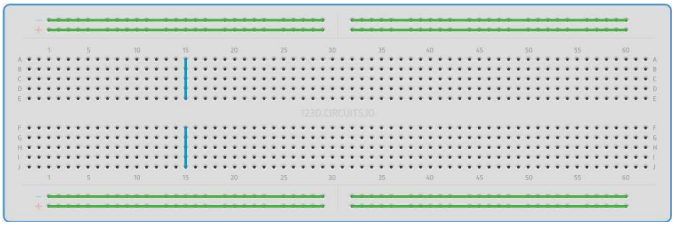

Fig. 1.1

1.2 Powering the Display

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.

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

The Arduino Uno has some ground pins, analog input pins A0-A3 and digital pins D1-D13 that can be used for both input as well as output. It also has two power pins that can generate $3.3V$ and $5V$. In the following exercises, only the GND, $5V$ and digital pins will be used.

Problem 1.3. Connect the 5V pin of the arduino to

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

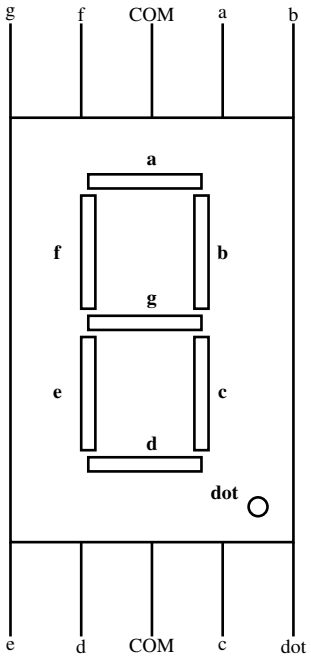

Fig. 1.2

an extreme pin that is in the same segment as the 1K resistor pin.

Problem 1.4. Connect the GND pin of the arduino to the opposite extreme pin of the breadboard

Problem 1.5. Connect the Arduino to the computer.

Problem 1.6. Connect the *dot pin of the display to a pin in the same segment as the GND pin. What do you observe?*

1.3 Controlling the Display

Fig. 1.9 explains how to get decimal digits using the seven segment display.

Problem 1.7. Generate the number 1 on the display by connecting only the pins b and c to GND.

Problem 1.8. Repeat the above exercise to generate the number 2 on the display.

Problem 1.9. Table 1.9 summarizes the process of generating the decimal digits. 0 means connecting to ground and 1 means not connecting. Complete Table 1.9 for all numbers between 0-9.

Problem 1.10. Now generate all numbers between 0-9 on the display using the above table.

The 7447 IC helps in displaying decimal numbers on the seven segment display. The $\bar{a} - \bar{g}$, pins of

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 1.9

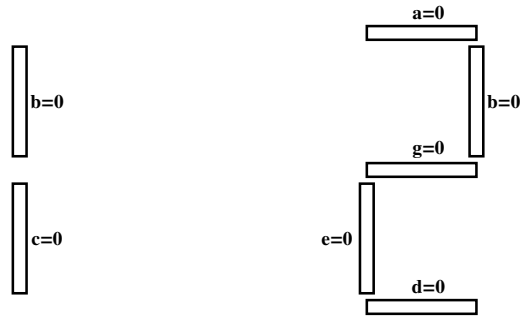

Fig. 1.9

the 7447 IC are connected to the $a - g$ 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

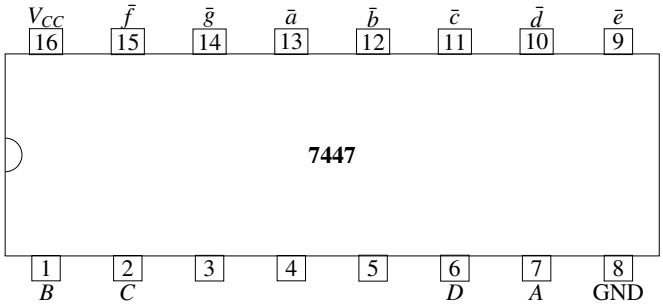

Fig. 1.10

Problem 1.11. Connect the 7447 IC decoder $\bar{a} - \bar{g}$ pins to the $a - g$ pins of the display respectively.

Problem 1.12. Connect the V_{cc} and GND pins of the decoder to the 5V supply and GND pins of the breadboard.

Problem 1.13. Connect the A,B,C,D pins to pins in the GND extreme segment of the breadboard. What do you observe.

Problem 1.14. Now remove the D pin from the breadboard and observe the display output.

Problem 1.15. Generate a table with A,B,C,D inputs and the equivalent decimal number output.

2 Display Control through Arduino Software

2.1 Driving the Segments

Open the arduino software. Check if the ports show Arduino Uno and click the appropriate button.

Problem 2.1. Connect the A-D pins of the 7447 IC to the pins D2-D5 of the Arduino.

Problem 2.2. Type the following code and execute. What do you observe?

```
//Code released under GNU GPL.
   Free to use for anything.
//Remove the following two lines
   if you are using the Arduino IDE

#include "Arduino.h"


// the setup function runs once
   when you press reset or power
   the board
void setup() {
        //declaring output pins
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
}

// the loop function runs over and
    over again forever
void loop() {

   //Writing data to the output
      pins
  digitalWrite(2, 1); //LSB
  digitalWrite(3, 0);
  digitalWrite(4, 1);
  digitalWrite(5, 0); //MSB
}
```

Problem 2.3. Now generate the numbers 0-9 by modifying the above program.

2.2 Counting Decoder

In the truth table in Table 2.6, W, X, Y, Z are the inputs and A, B, C, D are the outputs. This table represents the system that increments the numbers 0-8 by 1 and resets the number 9 to 0 Note that $D = 1$ for the inputs 0111 and 1000. Using *boolean* logic,

$$D = WXYZ' + W'X'Y'Z \qquad (2.3.1)$$

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

Problem 2.4. Write a program for implementing functions like (2.3.1) in Arduino for B,C and D.

Solution:

```
//Code released under GNU GPL.
   Free to use for anything.
//Remove the following  line if
   you are using the Arduino IDE
#include "Arduino.h"

//Declaring all variables as
   integers
int Z=0,Y=1,X=1,W=1;
int D,C,B,A;
// the setup function runs once
   when you press reset or power
   the board
void setup() {
        //declaring output pins
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
}

// the loop function runs over and
    over again forever
void loop() {
  A=0;
  B=(W&&!X&&!Y&&!Z) || (!W&&X&&!Y
     &&!Z) || (W&&!X&&Y&&!Z) || (!W
     &&X&&Y&&!Z);
  C=(W&&X&&!Y&&!Z) || (!W&&!X&&Y
     &&!Z) || (W&&!X&&Y&&!Z) || (!W
     &&X&&Y&&!Z);
  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 2.6

```
  //Writing data to the output
     pins
  digitalWrite(2, A); //LSB
  digitalWrite(3, B);
  digitalWrite(4, C);
  digitalWrite(5, D); //MSB
}
//&& is the AND operation
// || is the OR operation
// ! is the NOT operation
```

Problem 2.5. Modify the above program by changing W,X,Y,Z and A and execute. Verify that your results are consistent with Table 2.6.

Problem 2.6. Write the boolean logic function for A in terms of W, X, Y, Z.

Problem 2.7. Verify if your logic is correct by observing the output on the seven segment display.

Problem 2.8. Introduce delay and feedback in your code to realise a decade counter.

2.3 Display Decoder

Problem 2.9. 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 2.10. Obtain the logic functions for outputs a, b, c, d, e, f, g in terms of the inputs A, B, C, D.

Problem 2.11. Disconnect the arduino from IC 7447 and connect the pins D2-D8 in the Arduino directly to the seven segment display.

Problem 2.12. Write a new program to implement the logic in Problem 2.10 and observe the output in the display. You have designed the logic for IC 7447!

Problem 2.13. Now include your counting decoder program in the display decoder program and see if the display shows the consecutive number.

A decade counter counts the numbers from 0-9 and then resets to 0.

Problem 2.14. Suitably modify the above program to obtain a decade counter.

3 Decade Counter through Arduino

Open the blink program. You will see the following

```
//Code released under GNU GPL.
   Free to use for anything.
//Remove the following  line if
   you are using the Arduino IDE
#include "Arduino.h"

// the setup function runs once
   when you press reset or power
   the board
void setup() {
  // initialize digital pin 13 as
     an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and
    over again forever
void loop() {

// turn the LED on (HIGH is the
   voltage level)
  digitalWrite(13, 1);
// wait for a second
  delay(1000);
// turn the LED off by making the
   voltage LOW
  digitalWrite(13, 0);
// wait for a second
  delay(1000);
}
```

	INPUT				OUTPUT				CLOCK		5V			
	W	X	Y	Z	A	B	C	D						
Arduino	D6	D7	D8	D9	D2	D3	D4	D5	D13					
7474	5	9			2	12			CLK1	CLK2	1	4	10	13
7474			5	9			2	12	CLK1	CLK2	1	4	10	13
7447					7	1	2	6			16			

TABLE 3.3

Problem 3.1. Connect the digital pin 13 of the arduino to the *dot* pin of the display. Execute the Blink program.

Problem 3.2. Change the delay to 500 ms in the program and execute. What do you observe?

The 7474 IC in Fig. 3.2 has two D flip flops. The D pins denote the input and the Q pins denote the output. CLK denotes the clock input.

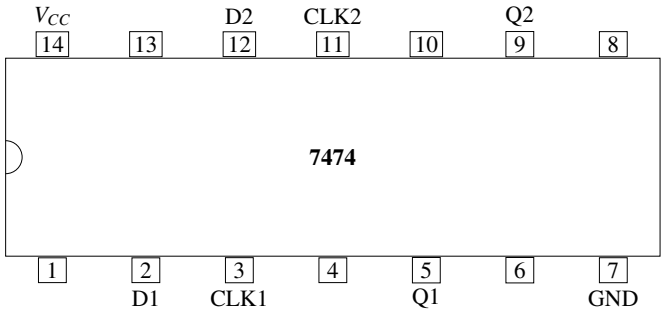

Fig. 3.2

Problem 3.3. Connect the Arduino, 7447 and the two 7474 ICs according to Table 3.3.

Problem 3.4. Connect the 7447 IC to the seven segment display.

Problem 3.5. Use the logic for the counting decoder in Problem 2.8 to implement the decade counter. You may refer to Fig. 3.5 to understand the functioning of a decade counter.

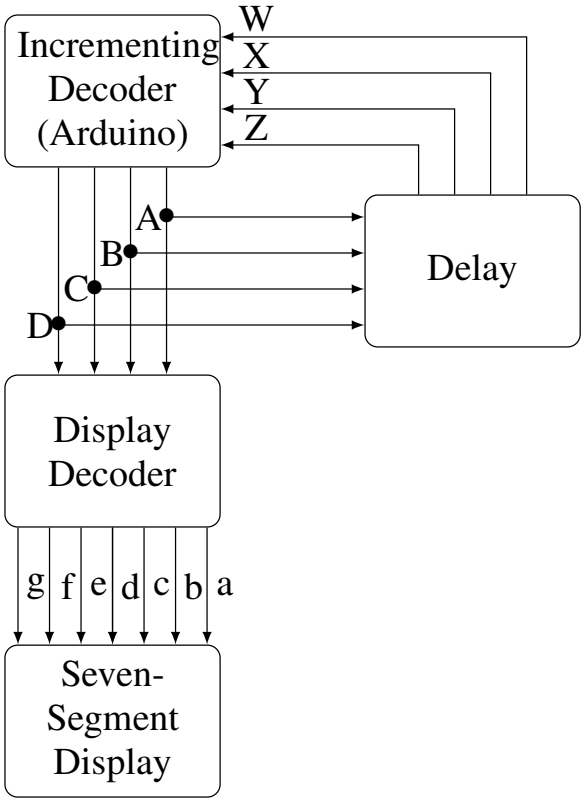

Fig. 3.5