C Programming through Arduino

G V V Sharma*

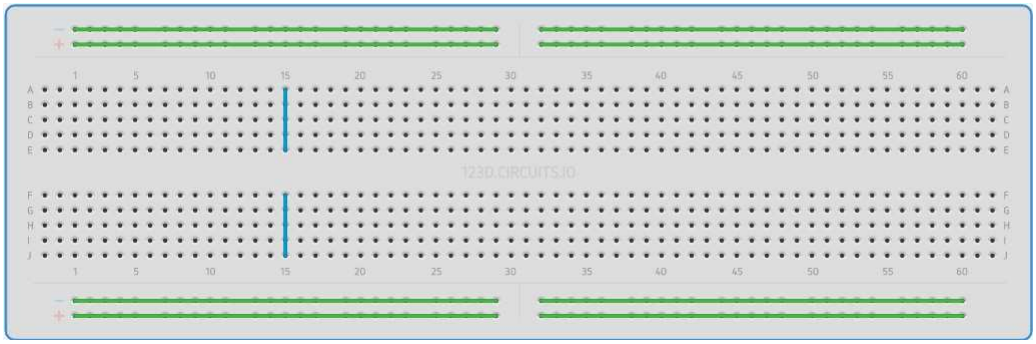
Contents

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

1 Display Control through Hardware

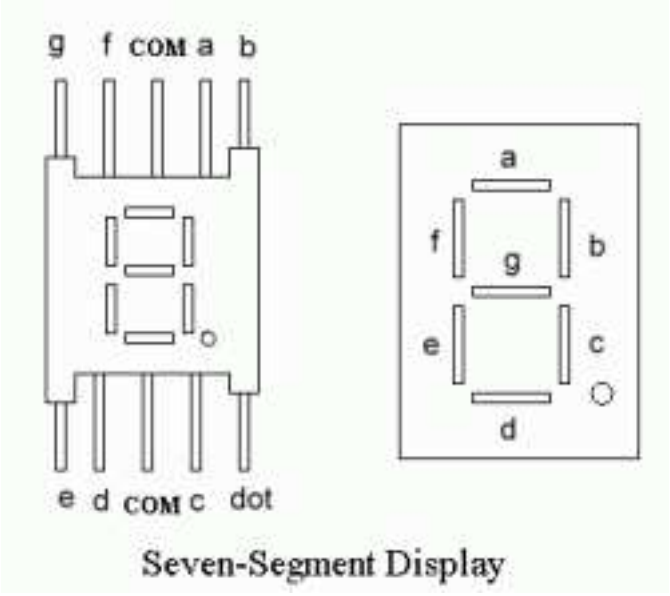
1.1 Powering the Display

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.*

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 1.2. *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 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 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?

1.2 Controlling the Display

Problem 1.7. *Generate the number 1 on the display by connecting the pins $a-g$ to GND according to the following table.*

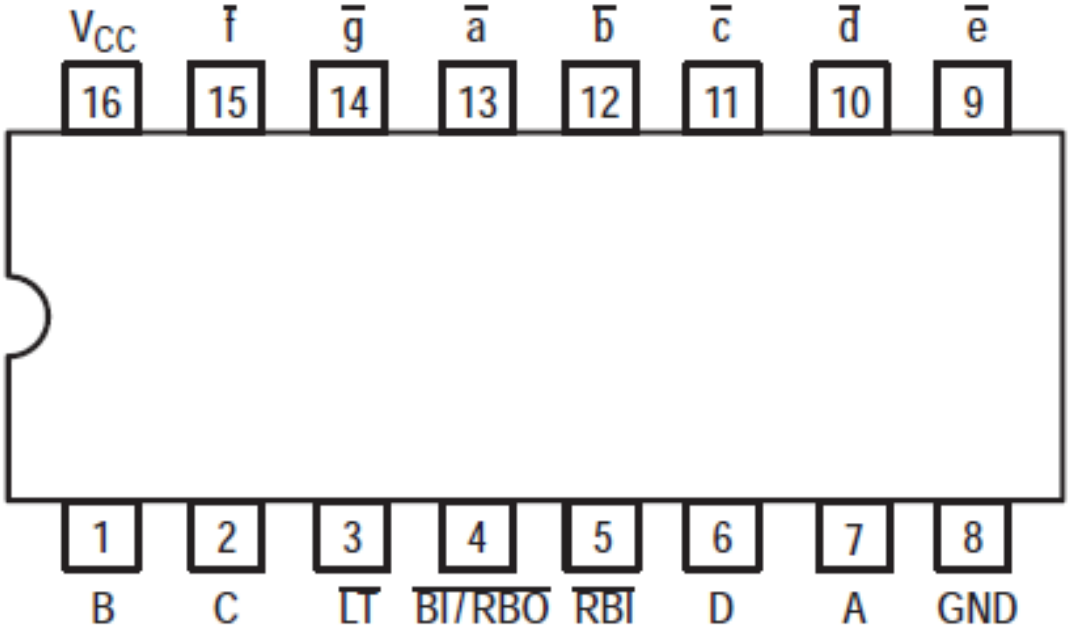
a	*b*	*c*	*d*	*e*	*f*	*g*	*decimal*
1	*0*	*0*	*1*	*1*	*1*	*1*	*1*

Problem 1.8. *Complete the above table for all numbers between 0-9.*

Problem 1.9. *Now generate the numbers from 1-9 on the display using the above table.*

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



Problem 1.10. *Connect the 7447 IC decoder $\bar{a} - \bar{g}$ pins to the $a - g$ pins of the display respectively.*

Problem 1.11. *Connect the V_{cc} and GND pins of the decoder to the 5V supply and GND pins of the breadboard.*

Problem 1.12. *Connect the A,B,C,D pins to pins in the GND extreme segment of the breadboard. What do you observe.*

Problem 1.13. *Now remove the D pin from the breadboard and observe the display output.*

Problem 1.14. *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. Open Examples→Basics→Blink.

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

Problem 2.2. *Modify the blink program to get the following code and execute. What do you observe?*

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

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

  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
}
```

Problem 2.3. *Now generate the numbers 1-9 by modifying the above program.*

Problem 2.4. *Before the following line in the previous code,*

```
void setup() {
```

you can define integer variables as

```
int A = 0;
```

where the variable A is defined to be an integer and given the values 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 1-9.

3 Combinational Logic

3.1 Counting Decoder

Following is a truth table, where 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.

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**

Note that $D = 1$ for the inputs 0111 and 1000. Using *boolean* logic,

$$D = WXYZ' + W'X'Y'Z \tag{3.1}$$

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

Problem 3.1. *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 (3.1) in Arudino is

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

Problem 3.2. *Write the Arduino code for the outputs A, B, C and verify if your logic is correct by observing the output on the seven segment display.*

3.2 Display Decoder

Problem 3.3. *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 3.4. *Obtain the logic functions for outputs a, b, c, d, e, f, g in terms of the inputs A, B, C, D.*

Problem 3.5. *Disconnect the arduino from IC 7447 and connect the pins D2-D8 in the Arduino directly to the seven segment display.*

Problem 3.6. *Write a new program to implement the logic in Problem 3.4 and observe the output in the display. You have designed the logic for IC 7447!*

Problem 3.7. *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 3.8. *Suitable modify the above program to obtain a decade counter.*

4 C Programming

Problem 4.1. *Write a module for decimal to binary conversion according to the example given below*

Decimal to Binary	*Example(N=9)*
A=N%2;	*A=9%2=1*
N=N/2;	*N=9/2=4*
B=N%2;	*B=4%2=0*
N=N/2;	*N=4/2=2*
C=N%2;	*C=2%2=0*
N=N/2;	*N=2/2=1*
D=N%2;	*D=1%2=1*
N=N/2;	*N=1/2=0*

N%2 gives the remainder and N/2 gives the quotient and use it in the above code so that decimal values are given as input in the program and observed as output in the display. Note that the following code

```
a % b
```

can be used to obtain the remainder when a is divided by b and

```
a/b
```

gives the quotient.

Problem 4.2. *Test the above module with both the counting decoder as well as the display decoder.*

Problem 4.3. *Use a* for *loop in C for the decimal to binary conversion.*

Problem 4.4. *Use the above* for *loop in a* function *and call this function in the main program.*

Problem 4.5. *Repeat the above exercises using a* while *loop*

Problem 4.6. *Instead of the counting decoder, use a* for/while *loop to implement the decade counter.*