**Interfacing 7 Segment Display to 8051**

Seven segment displays are used to indicate numerical information. Seven segments display can display digits from 0 to 9 and even we can display few characters like A, b, C, H, E, e, F, etc. These are very popular and have many more applications. So, in this project, I’ll show you how a 7 Segment Display works by interfacing 7 Segment Display to 8051 Microcontroller.

Before going to start this concept, get an idea about [**how to interface LEDs with 8051 Microcontroller**](https://www.electronicshub.org/led-interfacing-8051/).

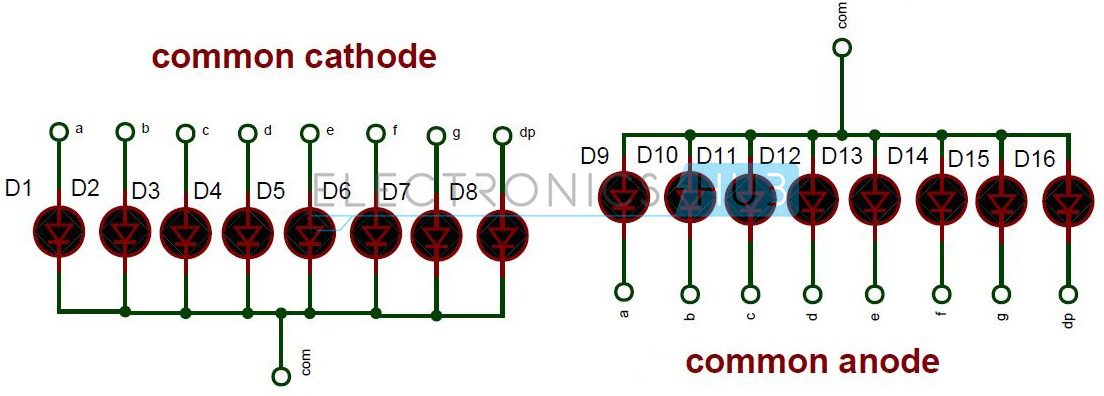
This article describes you how to interface seven segment display to AT89C51 microcontroller. This system displays the digits from 0 to 9 continuously with a predefined delay. In the process, I’ll design two circuits: one circuit with a single digit 7 Segment Display and the other circuit consists of a 4-digit 7 Segment Display.

Table of Contents

* [Circuit Principle](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_Principle)
* [Circuit Diagram](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_Diagram)
  + [Circuit 1: Interfacing 7 Segment Display to 8051 (Single Digit – CC)](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_1_Interfacing_7_Segment_Display_to_8051_Single_Digit_CC)
  + [Circuit 2: Interfacing 7 Segment Display to 8051 (4-Digit – CA)](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_2_Interfacing_7_Segment_Display_to_8051_4-Digit_CA)
* [Circuit Components](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_Components)
* [Circuit Design](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_Design)
  + [Circuit 1](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_1)
  + [Circuit 2](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Circuit_2)
* [Digit Drive Pattern](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Digit_Drive_Pattern)
* [Algorithm](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Algorithm)
  + [For Circuit 1](https://www.electronicshub.org/interfacing-7-segment-display-8051/#For_Circuit_1)
  + [For Circuit 2](https://www.electronicshub.org/interfacing-7-segment-display-8051/#For_Circuit_2)
* [CODE](https://www.electronicshub.org/interfacing-7-segment-display-8051/#CODE)
  + [Code for Circuit 1 (Single Digit)](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Code_for_Circuit_1_Single_Digit)
  + [Code for Circuit 2 (4-Digit)](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Code_for_Circuit_2_4-Digit)
* [Simulation Video of Circuit1](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Simulation_Video_of_Circuit1)
* [How to Operate?](https://www.electronicshub.org/interfacing-7-segment-display-8051/#How_to_Operate)
* [Project Output Video](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Project_Output_Video)
* [Applications](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Applications)
* [Limitations](https://www.electronicshub.org/interfacing-7-segment-display-8051/#Limitations)

**Circuit Principle**

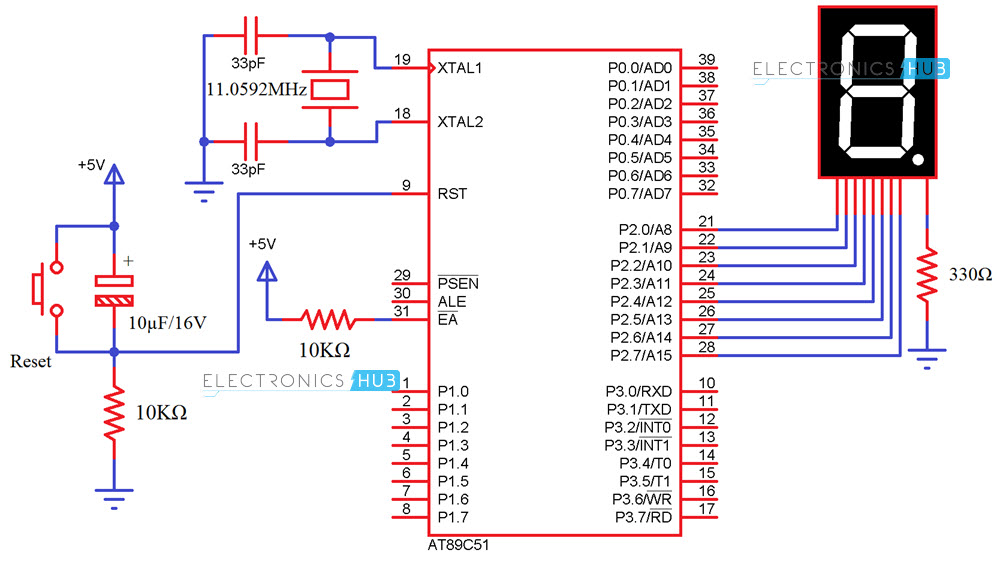
Seven segment displays internally consist of 8 LEDs. In these LEDs, 7 LEDs are used to indicate the digits 0 to 9 and single LED is used for indicating decimal point. Generally seven segments are two types, one is common cathode and the other is common anode.

In common cathode, all the cathodes of LEDs are tied together and labeled as com. and the anode are left alone. In common anode, seven segment display all the anodes are tied together and cathodes are left freely. Below figure shows the internal connections of seven segment Display. 

**Internal Connections of Seven Segment**

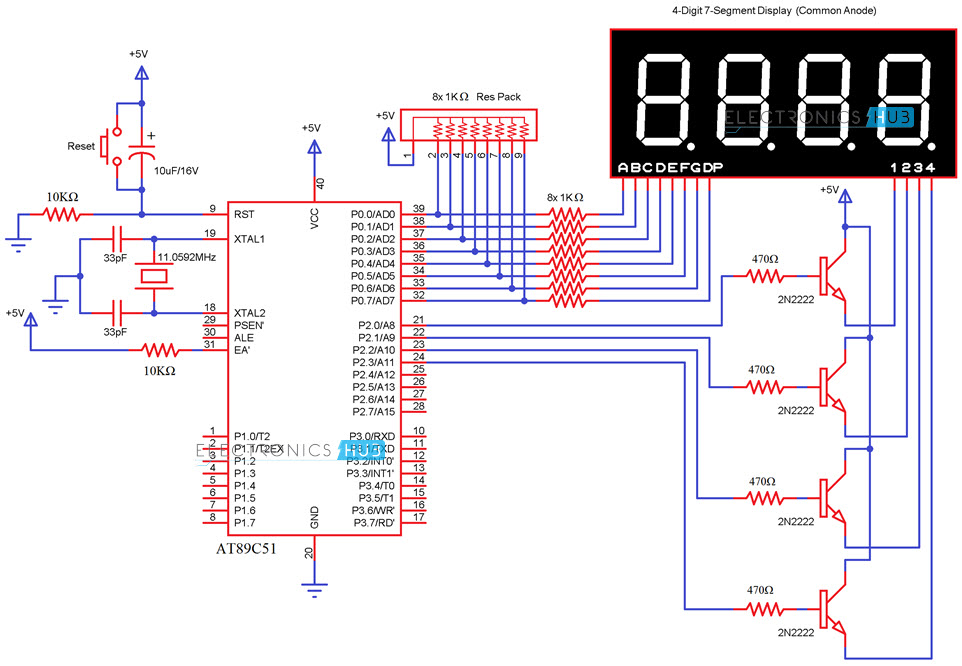
In the first circuit, I will interface a Common Cathode Single Digit 7 Segment Display to the 8051 Microcontroller while in the second circuit, I will interface a Common Anode type 4-Digit 7-Segment Display to the 8051 Microcontroller.

**Circuit Diagram**

**Circuit 1: Interfacing 7 Segment Display to 8051 (Single Digit – CC)** 

**Circuit 2: Interfacing 7 Segment Display to 8051 (4-Digit – CA)**

In the second circuit, I have interfaced a 4-digit 7-Segment Display to 8051 Microcontroller and the display type is of common anode.



**Circuit Components**

* AT89C51 Microcontroller
* AT89C51 Programming board
* Programming cable
* 12V DC battery or adaptor
* Common Cathode 7 segment Display
* Common Anode 4-Digit 7-Segment Display
* Resistors – 10KΩ X 2, 330Ω, 1KΩ X 8, 470Ω X 4
* 1KΩ X 8 Resistor Pack
* 33pF Ceramic capacitors x 2
* 11.0592 MHz crystal
* 10μF Electrolytic capacitor
* 2N2222 NPN Transistor X 4
* Push button
* Connecting wires

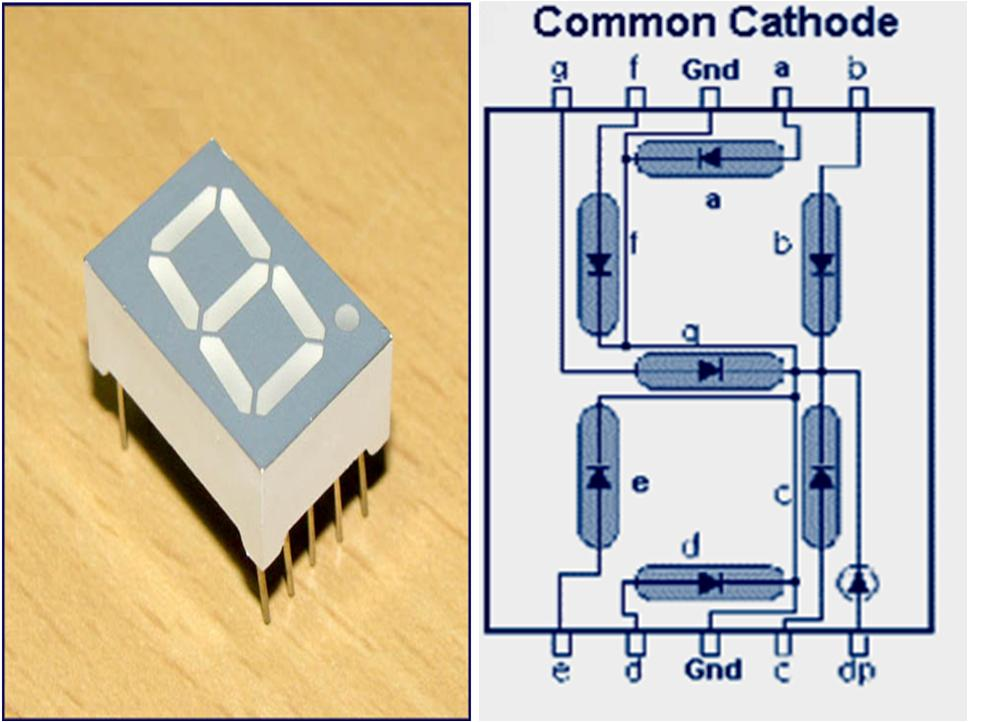
Also read the post –[[**How to Interface DC Motor with 8051 Microcontroller**](https://www.electronicshub.org/interfacing-dc-motor-8051-microcontroller/)**?**]

**Circuit Design**

**Circuit 1**

Here, common cathode seven segment is used to display the digits. In this circuit, pins a to h of the 7 segment are connected to the PORT 2 of the microcontroller and *com* pin is connected to the ground through the 330 ohm resistor. This resistor is used to drop the voltage. Since we are using common cathode seven segment we need to send LOGIC 1 to the segments to glow.

Figure shows structure of common cathode seven segments. Here dot is used for indicating the decimal point. Here all the cathodes of LED’s are connected to the Gnd pin. The operating voltage of this LED’s is 2 to 3V but from controller we will get 5V so to drop the remaining voltage we have to connect a to g pins to the controller through the resistor.



**Common cathode 7 segment Display**

**Circuit 2**

Since the 4-digit 7-Segment display used in the second circuit is of common anode type, we need to drive the LED segments through the common terminals. I have used 4 NPN Transistors to drive the 4 common anodes and the transistors are controlled by the 8051.

Coming to the segments a to h, they are connected to PORT0 Pins of 8051.

**Digit Drive Pattern**

To display the digits on 7 segment, we need to glow different logic combinations of segments. For example if you want to display the digit 3 on seven segment then you need to glow the segments a, b, c, d and g. The below table show you the Hex decimal values what we need to send from PORT2 to Display the digits from 0 to 9.

| **Digit** | **Dp** | **g** | **f** | **e** | **d** | **c** | **b** | **a** | **Hex value** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0x3f |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0x06 |
| 2 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0x5b |
| 3 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0x4f |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0x66 |
| 5 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0x6d |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0x7d |
| 7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0x07 |
| 8 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0x7f |
| 9 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0x67 |

**NOTE:** These values are suitable only for a Common Cathode display. If you want to drive a Common Anode display, then you have to take the complement of each bit and replace the hexadecimal values in the code (which I have done in the code of the second circuit).

**Algorithm**

**For Circuit 1**

1. First initialize all the segment hex values of the digits in an array.

unsigned char arr[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67};

1. Now take for loop and assign array values to the PORT2 with some time delay.

for (i=0;i<10;i++)

                        {

                                    P2=arr[i];

                                    delay\_ms(500);

                        }

**For Circuit 2**

1. First initialize all the segment hex values of the digits in an array.

unsigned char ch[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90}

       2. As per the value, switch the digits.

void display (unsigned long int n)

      {

          led=ch[n/1000];

          sw1=1;

          sdelay(30);

          sw1=0;

          led=ch[(n/100)%10];

          sw2=1;

          sdelay(30);

          sw2=0;

          led=ch[(n/10)%10];

          sw3=1;

          sdelay(30);

          sw3=0;

          led=ch[n%10];

          sw4=1;

          sdelay(30);

          sw4=0;

     }

**CODE**

**Code for Circuit 1 (Single Digit)**

The code and the simulation files for the first circuit can be downloaded from this link: [**Download Project Code**](https://www.electronicshub.org/wp-content/uploads/2015/12/Interfacing-7-segment-display-with-8051.rar)

**Code for Circuit 2 (4-Digit)**

|  |
| --- |
|  |
| #include<reg51.h> | |
|  | |

|  |
| --- |
| #define led P0 |
|  |

|  |
| --- |
| sbit sw1=P2^0; |
|  |

|  |
| --- |
| sbit sw2=P2^1; |
|  |

|  |
| --- |
| sbit sw3=P2^2; |
|  |

|  |
| --- |
| sbit sw4=P2^3; |
|  |

|  |
| --- |
| unsigned char ch[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90}; |
|  |

|  |
| --- |
| //void delay (int); |
|  |

|  |
| --- |
| void display (unsigned long int); |
|  |

|  |
| --- |
| void sdelay (char); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*void delay (int d) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| unsigned char i; |
|  |

|  |
| --- |
| for(;d>0;d--) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| for(i=250;i>0;i--); |
|  |

|  |
| --- |
| for(i=248;i>0;i--); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void sdelay (char d) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| for(;d>0;d--); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void display (unsigned long int n) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| led=ch[n/1000]; |
|  |

|  |
| --- |
| sw1=1; |
|  |

|  |
| --- |
| sdelay(30); |
|  |

|  |
| --- |
| sw1=0; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| led=ch[(n/100)%10]; |
|  |

|  |
| --- |
| sw2=1; |
|  |

|  |
| --- |
| sdelay(30); |
|  |

|  |
| --- |
| sw2=0; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| led=ch[(n/10)%10]; |
|  |

|  |
| --- |
| sw3=1; |
|  |

|  |
| --- |
| sdelay(30); |
|  |

|  |
| --- |
| sw3=0; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| led=ch[n%10]; |
|  |

|  |
| --- |
| sw4=1; |
|  |

|  |
| --- |
| sdelay(30); |
|  |

|  |
| --- |
| sw4=0; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void main() |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| unsigned long int i=1; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| while(1) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| i=++i; |
|  |

|  |
| --- |
| display(i); |
|  |

|  |
| --- |
| if(i==9999) |
|  |

|  |
| --- |
| i=0; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |

[view raw](https://gist.github.com/elktros/80c5aefc16a9cb68767d3d9393c96945/raw/bf3ee9f9c341d6af553aa336ba88f10b2c63f7db/8051_7_Segment_Interface_4_Digit.c) [8051\_7\_Segment\_Interface\_4\_Digit.c](https://gist.github.com/elktros/80c5aefc16a9cb68767d3d9393c96945#file-8051_7_segment_interface_4_digit-c) hosted with by [GitHub](https://github.com)

**Simulation Video of Circuit1**

**How to Operate?**

1. Initially burn the program to the microcontroller
2. Give the connections as per the circuit diagram
3. Make sure that a to g pins of 7 segment are connected to the P2.0 to P2.6 respectively in the first circuit and to PORT0 pins in the second circuit.
4. Switch on the supply, you can observe that digits 0 to 9 will display continuously with some delay in the first circuit and digits from 0 to 9999 will display on 4-digit 7-segment display in the second circuit.
5. Switch of the supply.

**Project Output Video**

**Applications**

* Seven segments are widely used in digital clocks to display the time.
* These are used in electronic meters for displaying the numerical information.
* Used in Instrument panels
* Used in digital readout displays.

**Limitations**

* The complexity is increased to display large information.
* It is not possible to display the symbols on seven segment.