

# \*\*\*\*\*Scalable Seven Segment Library V\_1.0\*\*\*\*\*

This library has been developed at CEDT, NSIT for the easy handling of any number of Seven Segment displays connected with Arduino in any plausible fashion which could be either common anode or common cathode later of which is demonstrated in the undermentioned figure(Fig 1).

## Hardware Requirements:

- 1)Connect your Digital pins to the seven segment by deciding the number of seven segment you want to display. For example here(in the undermentioned example) we want 4 SSD's so we connect 4 digital pins to arduino(here 10-13).Then Decide upon whether you wish for a decimal point in your program , if you do then connect 8 pins to the digital pins of arduino, else connect 7. These Selection give you a flexible option of adding any number of Seven Segments and mentioning the same in the setup() of the software.
- 2)Common pins are connected via NPN/PNP transistors to increase the current carrying capacity.
- 3)The segment pins are connected to the Digital pins vider the current limiting resistors.

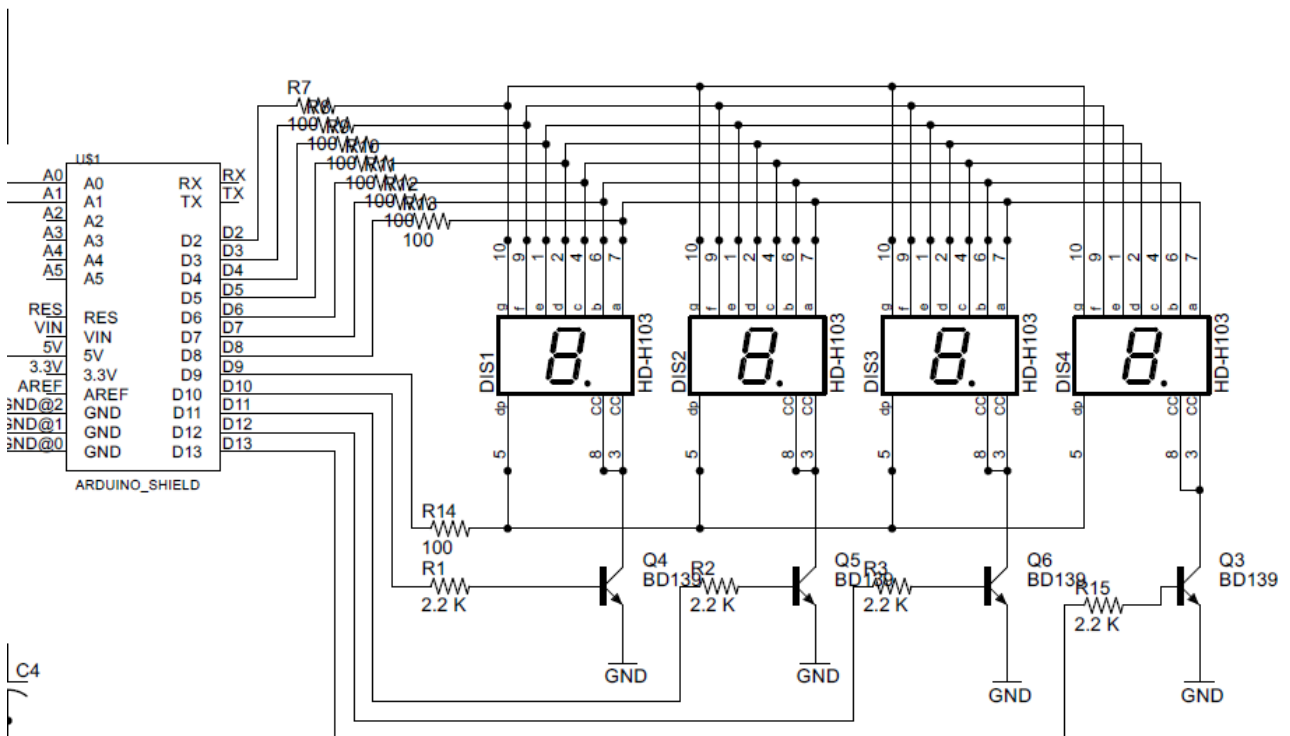


Fig 1 .Four Seven Segments connected in common cathode fashion.

## **Software :**

### **INSTRUCTIONS TO SET UP THE LIBRARY :**

- 1) Download the folder from the Git Repository and then copy the same in the libraries section of your Arduino folder to get the library working.
- 2) The folder also comes with an example folder containing a sample sketch demonstrating the syntactical use of the library.

### **What All Can This Library Do:**

- 1) This Library supports any number of seven Segments upto eleven the maximum an Arduino can support. (NOTE :Maximum : 11 in case of non decimal point requirements , And 10 in case of Decimal Point)
- 2) Each of the Seven Segment Display can display any of the hexadecimal character including an option to leave any of the Seven Segment Displayed (Referred to as the SSD hereafter), as blank.
- 3) The library call can handle "Raw Data" as well for example if you want to send "a" as a parameter then you just need to send 'a' and the library at the backend can handle it by itself.
- 4) Besides handling characters, library can also handle integer variables(Helpful for designing counter like stuff).
- 5) To leave any of the SSD blank you just need to pass 'g' or 16[Integer] as a parameter and that makes the SSD to shut off display.

### **SOFTWARE EXPLANATION :**

Although example code included in the repository is self explanatory , yet we describe the functions with there parameters and there role.

- 1) void Setup(boolean mode\_in, byte D1, byte D2, byte D3, byte D4, byte S1, byte S2, byte S3, byte S4, byte S5, byte S6, byte S7, byte S8) :

This function is always called in the beginning preferably in the setup() of the sketch to let library know the pin mapping of your hardware created .

NOTE : The parameters for this function can be variable in number to handle according to the users requirements.

- a)The first parameter is used to change the mode : 0 for Common Cathode and 1 for Common Anode.
  - b)The Second is used to Describe the number of Seven Segments you want to connect.
  - c)Third Parameter is a control of if you wish for a decimal point in the display or not.(1: Allow ; 0: Disallow)
  - c)Fourth Onwards uptill the number of seven segments you wish to connect describe the pin number of the common connections.
  - d)The Remaining describe the Seven Segment Pins Connections.
- 2) void Sendnumber(char number1,char number2,char number3,char number4, int DecPlace\_in) :

This function is used to update the number to be displayed.  
Note : This function has variable number of parameters depending upon the requirement.

- 3) void Display();

This functions makes the final formality of displaying the number with whatever has been updated in the Sendnumber parameter.

TO REM : By default the number is initialized to 0000 so when no number is updated and display() is called that makes it to display 0000 on the Four SSD's , provided you send the same in parameters of setup() .

#### Acknowledgments:

ScalableSSD Library  
Written by Rohan Seth, 2013

Version 1.0;

Direct any questions or suggestions to [rohanscar@gmail.com](mailto:rohanscar@gmail.com)

Special thanks to Prof Dhananjay V. Gadre, Co-ordinator CEDT,NSIT and Rohit Dureja (Undergraduate Student, NSIT ) for their continuous guidance and support during the development of the library.

Thanks to Dean Readings , Mark Chambers and Nathan Seidle for help used from their library SevenSeg,2013 to code this segment.