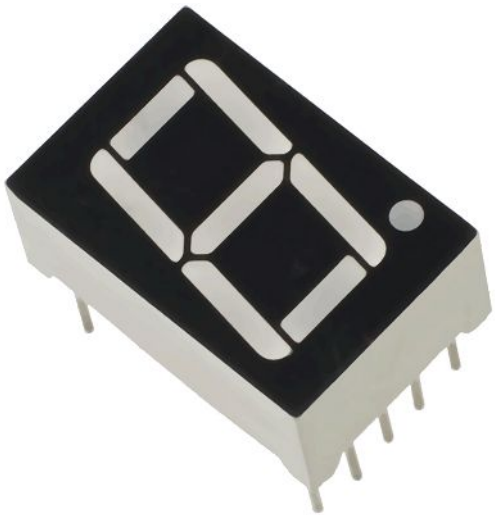
7-Segment Display Interfacing with Arduino UNO

Overview of 7-segment Display



7-Segment Display

7-segment displays are made up of 8 LED segments. They are used to display Numbers (0-9) and certain Alphabets (like c, A, H, P, etc.).

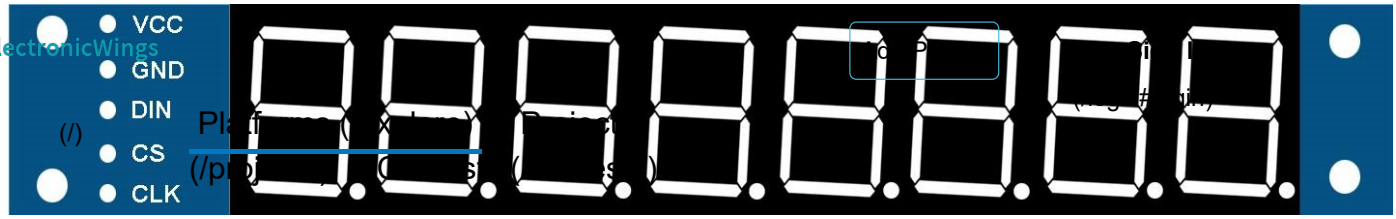
- 7 of these LED segments are in the shape of a line, whereas 1 segment is circular.
- Each of the 8 elements has a pin associated with it which can be driven HIGH or LOW.
- To display a number or alphabet, we need to turn on specific LED segments of the display.

For more information about 7-segment LED display and how to use it, refer the topic **7-segment LED Display** (http://www.electronicwings.com/sensors-modules/7-segment-led-display) in the sensors and modules section.

7-segment display module

There are various display modules available in the market that make use of ICs like the MAX7219 to drive multiple 7-segment displays using SPI communication. One such module is shown in the image given below.

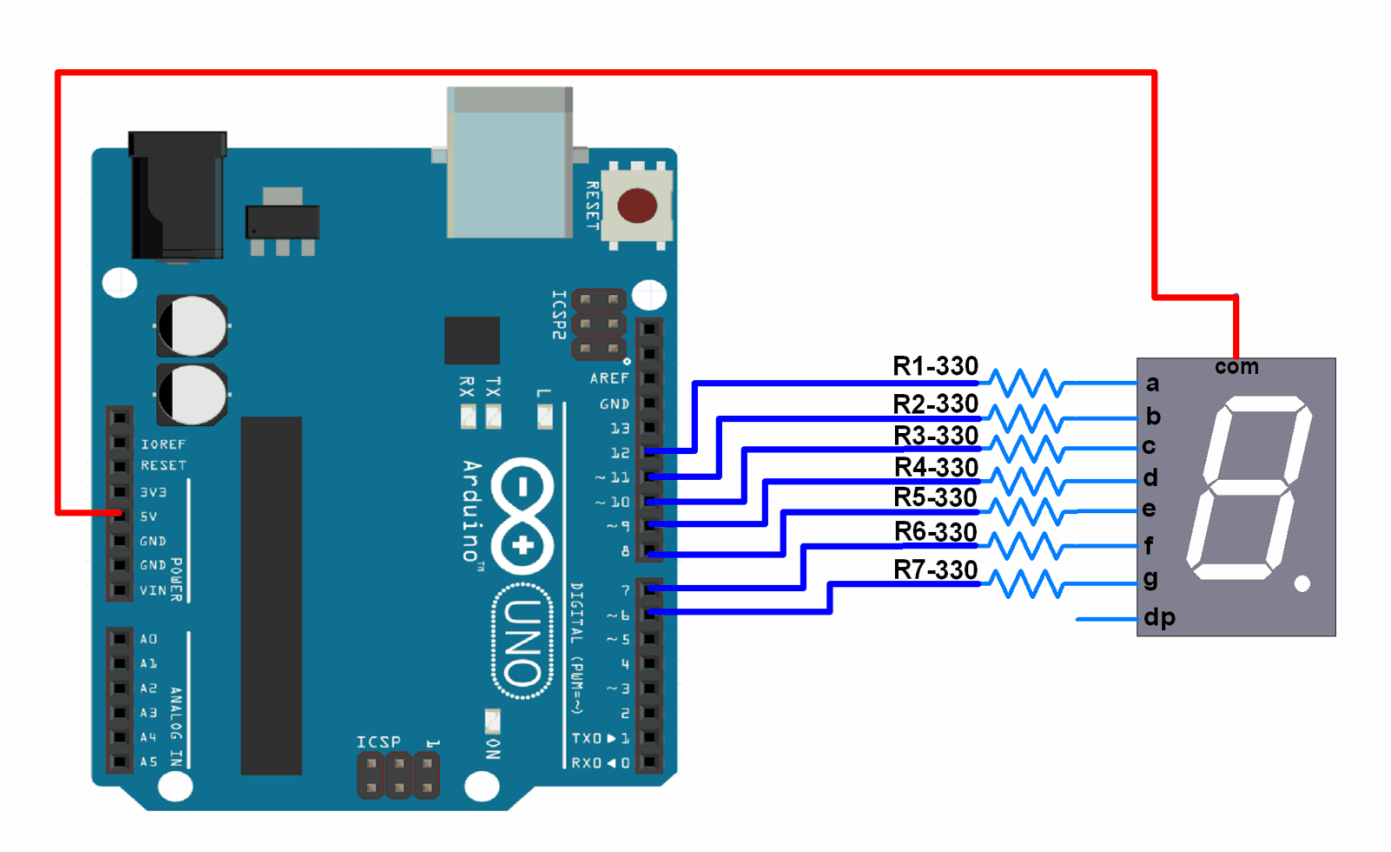
These modules are compact and require few pins and wires compared to using singular 7-segment displays in cascade.



7-Segment Display Module

We will be seeing how to interface both, a single 7-segment display as well as 7-segment display module that has eight 7-segment displays on it.

Connection Diagram of 7-Segment Display with Arduino UNO



Interfacing 7-Segment Display with Arduino UNO

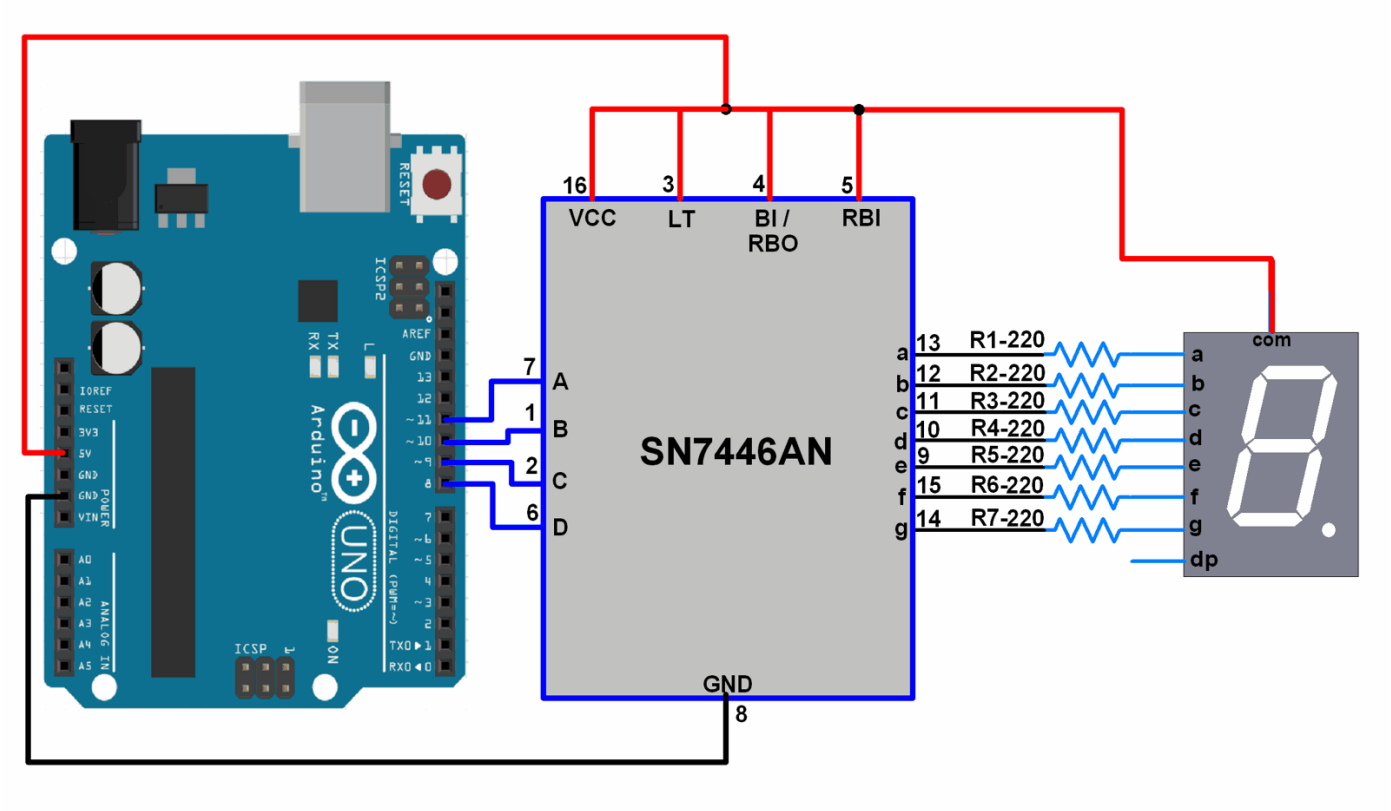
Here, the 7-Segment display is driven directly by Arduino. Resistors need to be connected between the display and the Arduino UNO board. Depending on which number or alphabet is to be displayed, control signals are applied.

> **Note:** We have used common anode display, hence the common pin is connected to 5V. If common cathode display is used, the common pin needs to be connected to ground.

For common anode display, drive pin LOW to turn on corresponding LED segment.

For common cathode display, drive pin HIGH to turn on the corresponding LED segment.

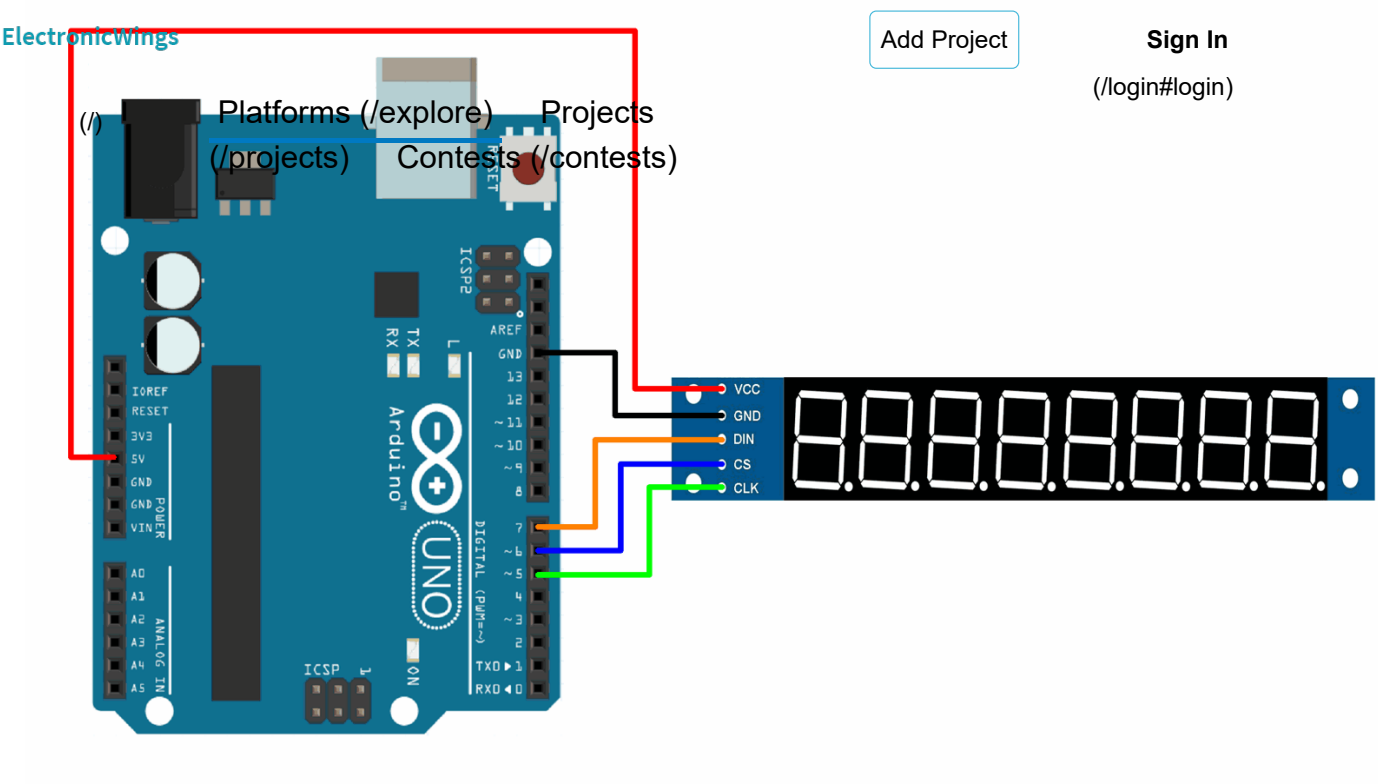
Connection Diagram of 7-Segment Display Using SN7446AN Driver with Arduino UNO



Interfacing 7-Segment Display Using SN7446AN Driver with Arduino UNO

Here, the 7-segment display is driven by the SN7446AN IC. It is a BCD to 7-segment driver/decoder IC. This reduces the number of pins required to drive the 7-segment display.

Connection Diagram 7-Segment Display Module with Arduino UNO



Interfacing 7-Segment Display Module with Arduino UNO

Here, we will be using the LED Control library by Wayoda from GitHub.

Using this library, it is possible to control up to eight SPI based 7-Segment modules (each module can have up to eight 7-segments displays) connected in cascade, using just 3 pins.

Download this library from here (https://github.com/wayoda/LedControl/archive/master.zip).

Extract the library and add it to the libraries folder path of Arduino IDE.

For information about how to add a custom library to the Arduino IDE and use examples from it, refer Adding Library to Arduino IDE (http://www.electronicwings.com/arduino/adding-a-new-library-to-arduino-ide-and-using-it) in the Basics section.

There are three example sketches available in the library.

We have created simplified sketch using the header file of this library.

7-Segment Display Code for Arduino

ElectronicWings Add Project Sign In (/login#login)
(/) Platforms (/explore) Projects (/projects) Contests (/contests)

```
int disp_pin[7]; /* array for a-g pins of 7-Segment display */
void define_segment_pins(int a, int b, int c, int d, int e, int f, int g)  /* Assigns 7-segment display pins to board
{
  disp_pin[0] = a;
  disp_pin[1] = b;
  disp_pin[2] = c;
  disp_pin[3] = d;
  disp_pin[4] = e;
  disp_pin[5] = f;
  disp_pin[6] = g;
}

void display_number(int num)          /* Function for displaying number (0-9) */
{
  switch(num)
  {
    case 0:
    digitalWrite(disp_pin[0], LOW);      /* Drive disp_pin[0] to LOW */
    digitalWrite(disp_pin[1], LOW);      /* Driving LOW turns on LED segment for common anode disp
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], LOW);
    digitalWrite(disp_pin[5], LOW);
    digitalWrite(disp_pin[6], HIGH);
    break;
    case 1:
    digitalWrite(disp_pin[0], HIGH);     /* Drive disp_pin[7] to HIGH */
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], HIGH);     /* Driving HIGH turns off LED segment for common anode dis
    digitalWrite(disp_pin[4], HIGH);
    digitalWrite(disp_pin[5], HIGH);
    digitalWrite(disp_pin[6], HIGH);
    break;
    case 2:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], HIGH);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], LOW);
    digitalWrite(disp_pin[5], HIGH);
    digitalWrite(disp_pin[6], LOW);
    break;
```

ElectronicWings (/) Platforms (/explore) Projects (/projects) Contests (/contests) Add Project Sign In (/login#login)

```
case 3:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], HIGH);
digitalWrite(disp_pin[6], LOW);
break;
case 4:
digitalWrite(disp_pin[0], HIGH);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], HIGH);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
case 5:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], HIGH);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
case 6:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], HIGH);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], LOW);
digitalWrite(disp_pin[4], LOW);
digitalWrite(disp_pin[5], LOW);
digitalWrite(disp_pin[6], LOW);
break;
case 7:
digitalWrite(disp_pin[0], LOW);
digitalWrite(disp_pin[1], LOW);
digitalWrite(disp_pin[2], LOW);
digitalWrite(disp_pin[3], HIGH);
digitalWrite(disp_pin[4], HIGH);
digitalWrite(disp_pin[5], HIGH);
digitalWrite(disp_pin[6], HIGH);
break;
```

ElectronicWings (/) Platforms (/explore) Projects (/projects) Contests (/contests) Add Project Sign In (/login#login)

```
    case 8:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], LOW);
    digitalWrite(disp_pin[5], LOW);
    digitalWrite(disp_pin[6], LOW);
    break;
    case 9:
    digitalWrite(disp_pin[0], LOW);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], HIGH);
    digitalWrite(disp_pin[5], LOW);
    digitalWrite(disp_pin[6], LOW);
    break;
    default:
    digitalWrite(disp_pin[0], HIGH);
    digitalWrite(disp_pin[1], LOW);
    digitalWrite(disp_pin[2], LOW);
    digitalWrite(disp_pin[3], LOW);
    digitalWrite(disp_pin[4], LOW);
    digitalWrite(disp_pin[5], HIGH);
    digitalWrite(disp_pin[6], LOW);
    break;
  }
}

void setup() {
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  define_segment_pins(12,11,10,9,8,7,6);        /* a-g segment pins to Arduino */
}

void loop() {
  int i;
  for(i = 9; i>=0; i--)
  {
    display_number(i);
```

```
    delay(1000);
  }
  for(i = 0; i<=9; i++)
  {
    display_number(i);
    delay(1000);
  }
}
```


Video of 7-Segment display using Arduino Uno

SN7446AN Decoder Code for Arduino

ElectronicWings Add Project Sign In (/login#login)
Platforms (/explore) Projects (/projects) Contests (/contests)

```
int bcd_pins[4];                    /* array for A-D pins of driver IC */

void bcd_control_pins(int a, int b, int c, int d)  /* Assigns A-D pins of deiver IC to Arduino board */
{
  bcd_pins[0] = a;
  bcd_pins[1] = b;
  bcd_pins[2] = c;
  bcd_pins[3] = d;
}

void display_number(int num)        /* Function for displaying number (0-9) */
{
  switch(num)
  {
    case 0:
    digitalWrite(bcd_pins[0], LOW);     /* Drive bcd_pin[0] to LOW */
    digitalWrite(bcd_pins[1], LOW);     /* Driving LOW turns on LED segment for common anode disp
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 1:
    digitalWrite(bcd_pins[0], HIGH);
    digitalWrite(bcd_pins[1], LOW);
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 2:
    digitalWrite(bcd_pins[0], LOW);
    digitalWrite(bcd_pins[1], HIGH);
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 3:
    digitalWrite(bcd_pins[0], HIGH);    /* Drive bcd_pin[3] to HIGH */
    digitalWrite(bcd_pins[1], HIGH);    /* Driving HIGH turns on LED segment for common anode disp
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 4:
    digitalWrite(bcd_pins[0], LOW);
    digitalWrite(bcd_pins[1], LOW);
    digitalWrite(bcd_pins[2], HIGH);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 5:
```

ElectronicWings (/) Platforms (/explore) Projects (/projects) Contests (/contests) Add Project Sign In (/login#login)

```
    digitalWrite(bcd_pins[0], HIGH);
    digitalWrite(bcd_pins[1], LOW);
    digitalWrite(bcd_pins[2], HIGH);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 6:
    digitalWrite(bcd_pins[0], LOW);
    digitalWrite(bcd_pins[1], HIGH);
    digitalWrite(bcd_pins[2], HIGH);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 7:
    digitalWrite(bcd_pins[0], HIGH);
    digitalWrite(bcd_pins[1], HIGH);
    digitalWrite(bcd_pins[2], HIGH);
    digitalWrite(bcd_pins[3], LOW);
    break;
    case 8:
    digitalWrite(bcd_pins[0], LOW);
    digitalWrite(bcd_pins[1], LOW);
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], HIGH);
    break;
    case 9:
    digitalWrite(bcd_pins[0], HIGH);
    digitalWrite(bcd_pins[1], LOW);
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], HIGH);
    break;
    default:
    digitalWrite(bcd_pins[0], LOW);
    digitalWrite(bcd_pins[1], LOW);
    digitalWrite(bcd_pins[2], LOW);
    digitalWrite(bcd_pins[3], LOW);
    break;
  }
}

void setup() {
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  bcd_control_pins(11,10,9,8);  /* A-D of driver IC to Arduino */
}
```

```
void loop() {
  int i;
  for(i = 9; i>=0; i--)
  {
    display_number(i);
    delay(1000);
  }
  for(i = 0; i<=9; i++)
  {
    display_number(i);
    delay(1000);
  }
}
```

ElectronicWings Platforms (/explore) Projects (/projects) Contests (/contests) Add Project Sign In (/login#login)

7-Segment Display Module Code for Arduino Uno

ElectronicWings Add Project Sign In (/login#login)
Platforms (/explore) Projects (/projects) Contests (/contests)

```
#include "LedControl.h"

LedControl new_disp = LedControl(7,5,6,1);      /* Create an LED control object */

void setup() {
  new_disp.shutdown(0,false);     /* Normal functioning of device device with address 0 */
  new_disp.setIntensity(0,8);     /* Define medium(8) intensity of display device with address 0 */
  new_disp.clearDisplay(0);               /* Clear display device with address 0 */
}

void loop() {
  new_disp.setDigit(0,0,0,false);         /* Display 0 on the 1st 7-seg display */
  delay(1000);
  new_disp.setDigit(0,1,2,false);
  delay(1000);
  new_disp.setDigit(0,2,4,false);
  delay(1000);
  new_disp.setDigit(0,3,6,false);
  delay(1000);
  new_disp.setChar(0,4,'F',false);        /* Display F on the 5 th 7-seg display */
  delay(1000);
  new_disp.setChar(0,5,'P',false);
  delay(1000);
  new_disp.setChar(0,6,'c',false);
  delay(1000);
  new_disp.setChar(0,7,'A',false);
  delay(1000);
  new_disp.clearDisplay(0);               /* Clear display device with address 0 */
  delay(1000);
}
```

Functions Used in Code

```
LedControl new_disp = LedControl(dataPin, clkPin,  csPin,  Devices)
```

- This creates an object of class LEDControl (here, new_disp; user can any other valid name as well) that talks to the MAX7219 on the 7-segment display module.
- **dataPin** is the pin of Arduino that is connected the DataIn pin on the module. This is the pin on the Arduino where data gets shifted out.
- **clkPin** is the pin of Arduino that is connected the CLK pin on the module. This is the pin for the clock.
- **csPin** is the pin of Arduino that is connected the CS/Load pin on the module.This is the pin for selecting the device when data is to be sent.
- **Devices** define the maximum number of devices that can be controlled. It can be between 1 to 8. It is the

ElectronicWings (/) Platforms (/explore) Projects (/projects) Contests (/contests) Add Project Sign In (/login#login)

number of devices that are connected in cascade.

- The device directly connected to the pins defined has address 0. The device immediately next to this in cascade has address 1. The 8 the device (last device in the cascade connection) has address 7.

```
new_disp.shutdown(addr, status)
```

- This function is used for power saving.
- If status is true, the device corresponding to addr goes into power down mode. If status is false, the device functions normally.

```
new_disp.setIntensity(addr, intensity)
```

- This function is used to set the brightness of the display whose address is addr.
- intensity can be between 1 to 15. 1 being minimum and 15 being maximum.

```
new_disp.setDigit(addr, digit, value, dp)
```

- This function is used for displaying a number (value) on the display with address addr. The number is displayed at the position(units, tens, etc.) mentioned by digit (can be 0 to 7).dp sets the decimal point. If dp is false, decimal point is off.

```
new_disp.setChar(addr, digit, value, dp)
```

- This function is used for displaying a character (value) on the display with address addr. The character is displayed at the position(units, tens, etc.) mentioned by digit (can be 0 to 7). dp sets the decimal point. If dp is false, decimal point is off.
- Only certain characters can be displayed.

Powered By MOUSER ELECTRONICS (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Components Used

Components Used

Arduino UNO
Arduino UNO

X 1

(https://www.mouser.in/ProductDetail/Arduino/A000073?qs=8PMfw1Pw72VfrrCu0MIn0mA%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (/components/arduino-uno-rev3/datasheet)

Arduino Nano
Arduino Nano

X 1

(https://www.mouser.com/ProductDetail/Arduino/ABX00033?qs=PzGy0jfpSMueXyfBfl2XSA%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (/components/arduino-nano/1/datasheet)

Components Used

Seven 7 Segment Display Seven 7 Segment Display	**X 1**	(https://www.mouser.in/ProductDetail/Vishay-Semiconductors/TDSR0760?qs=RzxYCzJDjPUsfuITQhx9uw%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/seven-7-segment-display/1/datasheet)
Breadboard Breadboard	**X 1**	(https://www.mouser.com/ProductDetail/BusBoard-Prototype-Systems/BB830?qs=VEfmQw3KOauhPeTwYxNCaA%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/breadboard/1/datasheet)

Supporting Files

Source Code

- 7_Segment_Interfacing_With_Arduino_INO Download | 632

Attached File

- LEDControl Library Download | 422

See also

- LedControl GitHub Page (https://github.com/wayoda/LedControl)

Comments

Please login/signup to comment and reply.

J

jeevankumarvatturi
(/users/jeevankumarvatturi/profile)
2019-11-21 10:44:40

Even though my circuit connections are perfectly right , in the output that means in the 7-segment it is displaying '0' continuously . If the input is changed output is not varying.
http://bigbelectronics.in/product.php?product=sn74ls47n-bcd-7-segment-decoder-driver

Reply Like