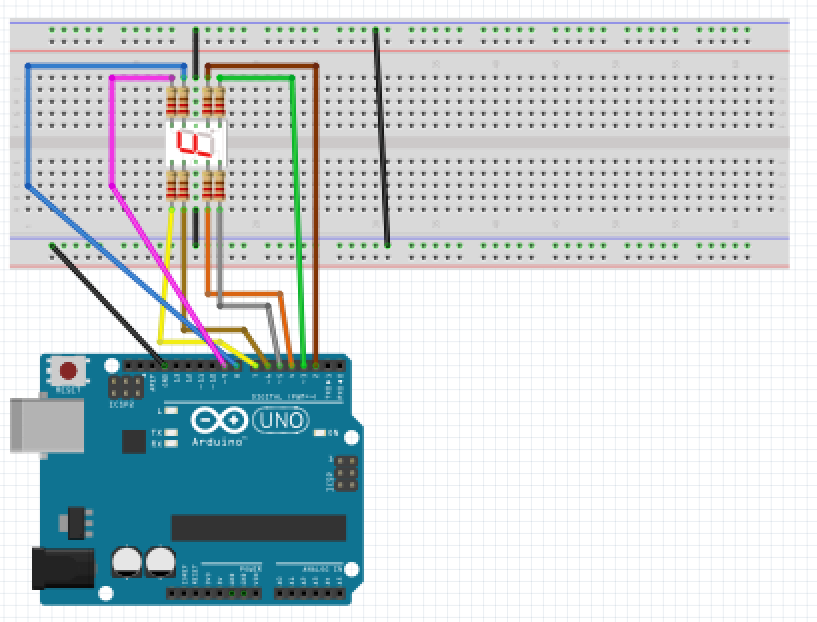
**ARDUINO LED PROJECT**

**7 Segment LED display.**



For this project we will need:

* Arduino board.
* 7 Segment LED Display
* Breadboard.
* 10 Male to Male jumper wires.
* 8 470 Ohm resistor

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the orientation of your 7 segment LED. The dot(.) indicates the right lower side.There are ten available pins attached.
3. Plug the 7 Segment LED to the breadboard.
4. Plug the third pin on either side the common ground of the breadboard using a black jumper wire. This are our common GND for the display.
5. Plug a 470 Ohm resistor to the corresponding line connected to 4th pin on the right .Using a brown jumper wire, plug the other resistor end to pin 2 of the Arduino.
6. Plug a 470 Ohm resistor to the corresponding line connected to 5th pin on the right. Using a green jumper wire, plug the other resistor end to pin 3 of the Arduino.
7. Plug a 470 Ohm resistor to the corresponding line connected to 4th pin on the left. Using a orange jumper wire, plug the other resistor end to pin 4 of the Arduino.
8. Plug a 470 Ohm resistor to the corresponding line connected to 5th pin on the left .Using a grey jumper wire, plug the other resistor end to pin 5 of the Arduino.
9. Plug a 470 Ohm resistor to the corresponding line connected to 2nd pin on the left. Using a ochre jumper wire, plug the other resistor end to pin 6 of the Arduino.
10. Plug a 470 Ohm resistor to the corresponding line connected to 1st pin on the left. Using a yellow jumper wire, plug the other resistor end to pin 7 of the Arduino.
11. Plug a 470 Ohm resistor to the corresponding line connected to 2nd pin on the right. Using a blue jumper wire, plug the other resistor end to pin 8 of the Arduino.
12. Plug a 470 Ohm resistor to the corresponding line connected to 1st pin on the right. Using a pink jumper wire, plug the other resistor end to pin 9 of the Arduino.
13. Plug the common ground of the breadboard to the Arduino using a black jumper wire.

|  |
| --- |
| const byte numeral[11]={  B11111100, //0  B01100000,  B11011010, //2  B11110010,  B01100110,//4  B10110110,  B00111110,//6  B11100000,  B11111110,//8  B11100110,  B00000000 // shows nothing  };  const int segmentPins[8]={5,8,9,7,6,4,3,2};  void setup(){  for (int i=0;i<=8;i++){  pinMode(segmentPins[i],OUTPUT);  }  }  void loop(){  for(int i= 0;i<=10;i++){  showDigit(i);  delay(1000); //duration to show specific digit  }  delay(2000); //delay how long to wait for the count to start again.  }  }  void showDigit(int number){  boolean isBitSet;  for(int segment =1 ; segment<8 ;segment++){  isBitSet=bitRead(numeral[number],segment);  digitalWrite(segmentPins[segment],isBitSet);  }  } |

const byte numera[11]={}

We declare the values we want to display. They are stored in a structured list known as array.

They are unique because they depend on indexes to access & store data.

This enables built in operations on the values stored in this data structure.

const the value cannot be changed during program execution or reassigned a new value.

byte the values stored are of binary type ie exist only in 2 states ON/OFF(TRUE/FALSE,1/0) .

B11111100

This 7 segment LED as the name suggests has 7 segments. It consists of 8 LEDS connected together. The additional LED shows the decimal point of digits if any.

In this case we are turning on LEDS for segments 1-6(HIGH/1).The seventh segment which is the center horizontal LED will not be lit. The last byte will also be OFF(LOW/0). Thus it will display 0.

The LEDs are numbered starting with the top horizontal as first, Then clockwise through the end. The middle horizontal one is the 7th LED , while decimal point(.) is the 8.

void showDigit(int number){

This is a function. It accepts a number. The number has to be of value integer.

boolean isBitSet;

This is a declaration statement.It reserves memory for a value which can only have one of two states.

It ensures the value is either TRUE/FALSE or in this case 1/0.

isBitSet=bitRead(numeral[number],segment);

This sets the value of isBitSet. This is achieved by checking the corresponding byte if its set.

If it has value of 1 its set to evaluate TRUE.

If it has a value of 0 it will evaluate as FALSE.

digitalWrite(segmentPins[segment],isBitSet);

Using the value set we can use to turn ON/OFF the associated segments LED.

If true it will turn on, if false it will turn off.

void setup (){}

This initializes the arduino and assigns functionality to its pins.

This also provides required resources for monitoring.

void loop(){}

After executing the void setup() function, we enter the void loop() and this function is executed continuously and repeatedly, until your Arduino is powered off.