184763_MUDATHIR EKUNDAYO_DOCUMENTATION

CIRCUIT DESCRIPTION

This circuit performs the basic operation of a calculator. It uses a keypad for inputs and three seven segments display to output the entered numbers. I used PORT D to connect my seven segment in order to allow for an easy display of the entered numbers using port manipulation techniques. Keypad rows were connected to PORT C (A0-A3) and the keypad columns were connected to PORT B (8-11). Choice of connection was to make the program easy and efficient to write using port manipulation. Also an overflow led was added and it blinks four times anytime the user inputs more than three digits.

Working principle of device:

INPUT OPERATION

User: enters 3

Seven segment: displays 3 on the rightmost seven segment

User: enters 5

Seven segment: displays 35; ‘3’ on the middle segment and ‘5’ on the rightmost segment

User: enters 7

Seven segment: displays 357; ‘3’ on the leftmost seven segment,’5’ on the middle segment and ‘7’on the rightmost segment.

User: enters 4

Seven segment: displays nothing and blinks overflow led 4 times, awaits new inputs

ADDITION, SUBTRACTION, DISPLAY AND RESET OPERATION

User: enters 3 and then 5

Seven segment: displays 35

User: enters A

Seven segment: clears the segment, current sum is 25 and allows for new input

User: enters 4 and then 7

Seven segment: displays 47

User: presses A

Seven segment: goes blank, adds 47 to 35

User: presses C

Seven segment: displays 82

User: enters 2 and then 5

Seven segment: displays 25

User: presses B

Seven segment: becomes blank, subtract 25 from 82

User: presses C

Seven segment: displays 57

User: presses D

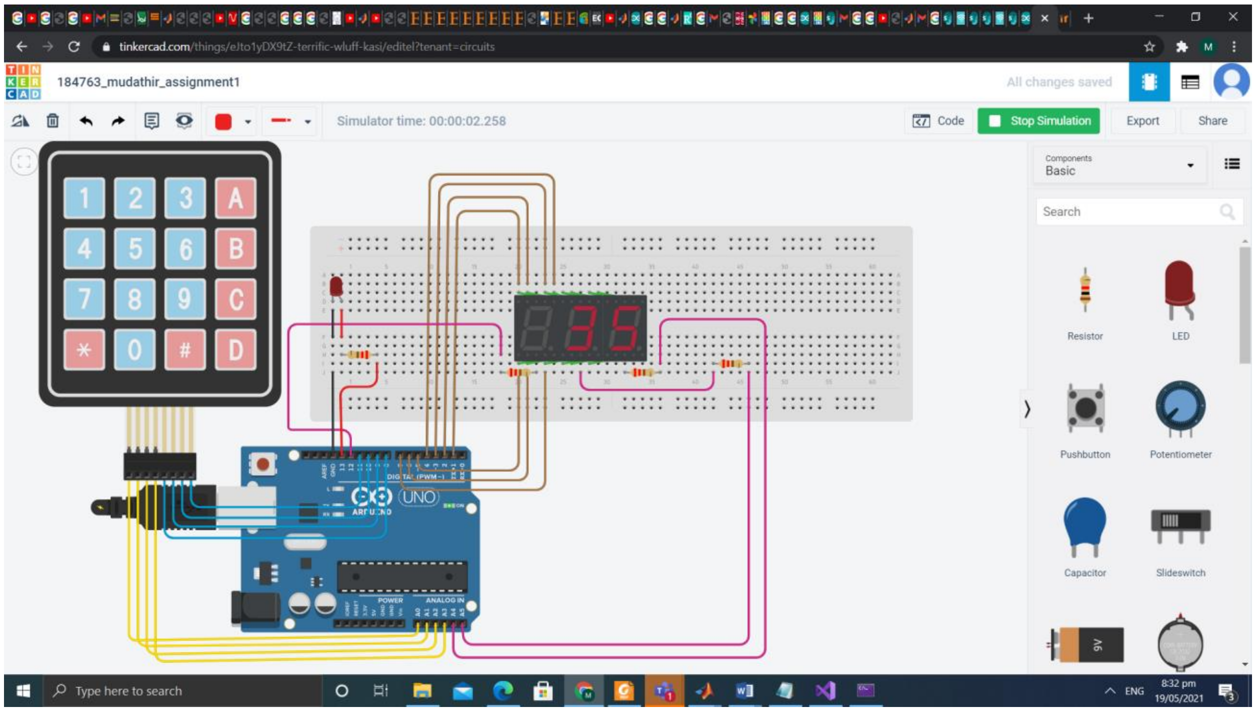
Seven segment: becomes blank

User: presses C

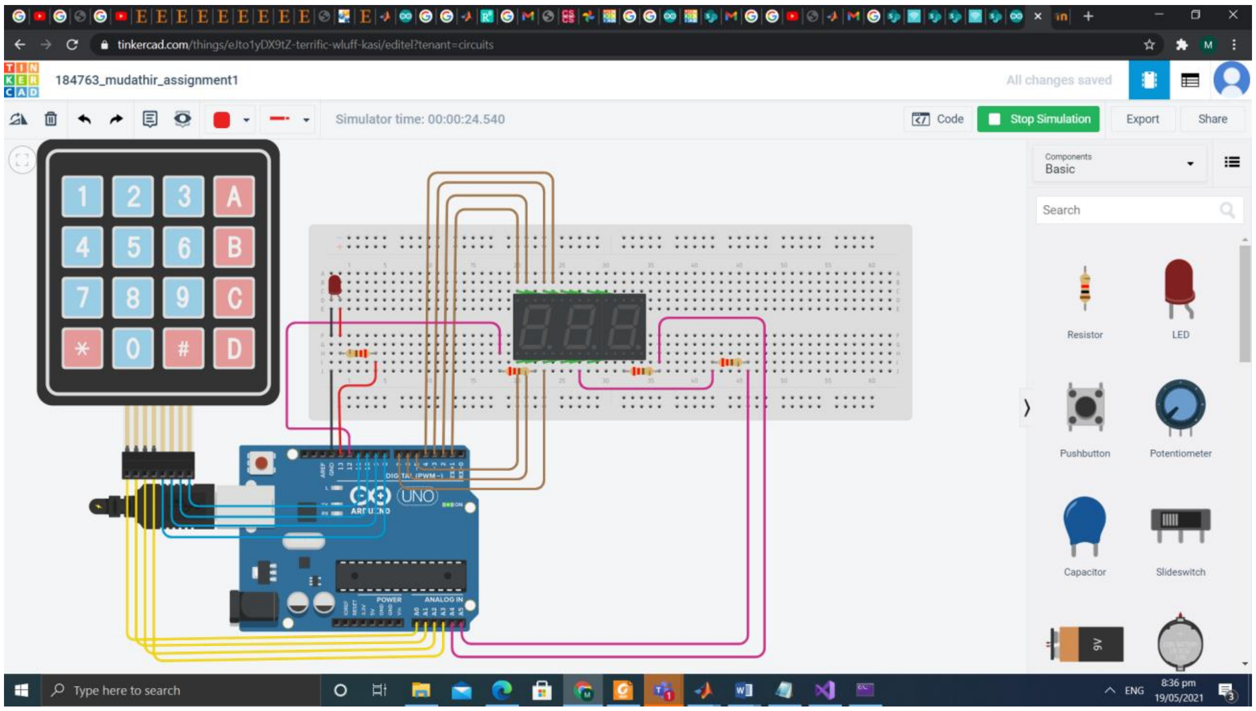
Seven segment: seven segment displays 0

SCREENSHOTS OF THE ABOVE WORKING PRINCIPLE

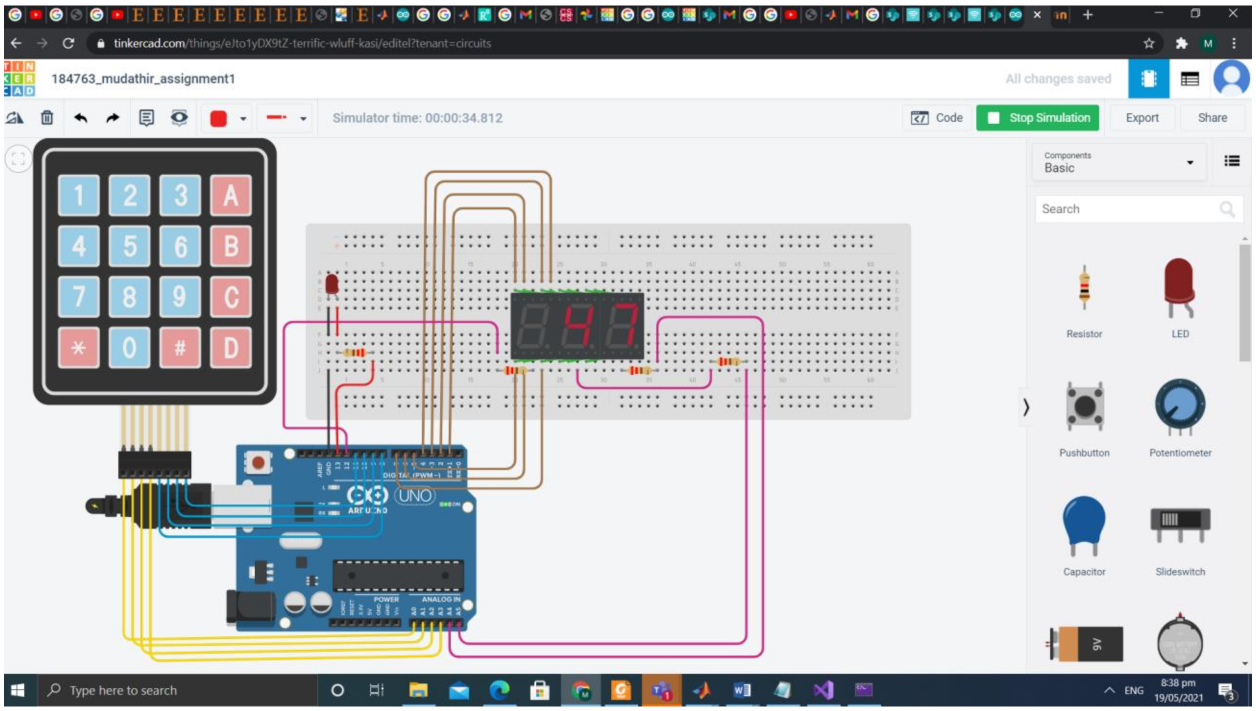
ENTERS 35



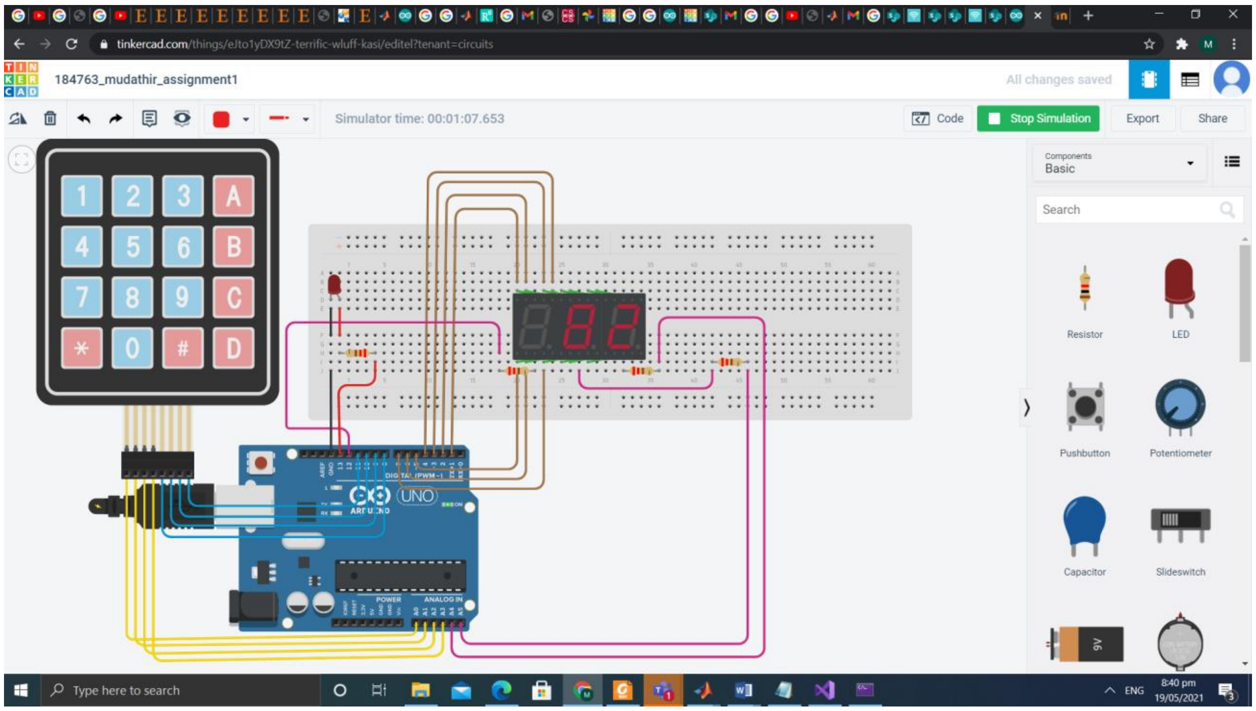
PRESSES A



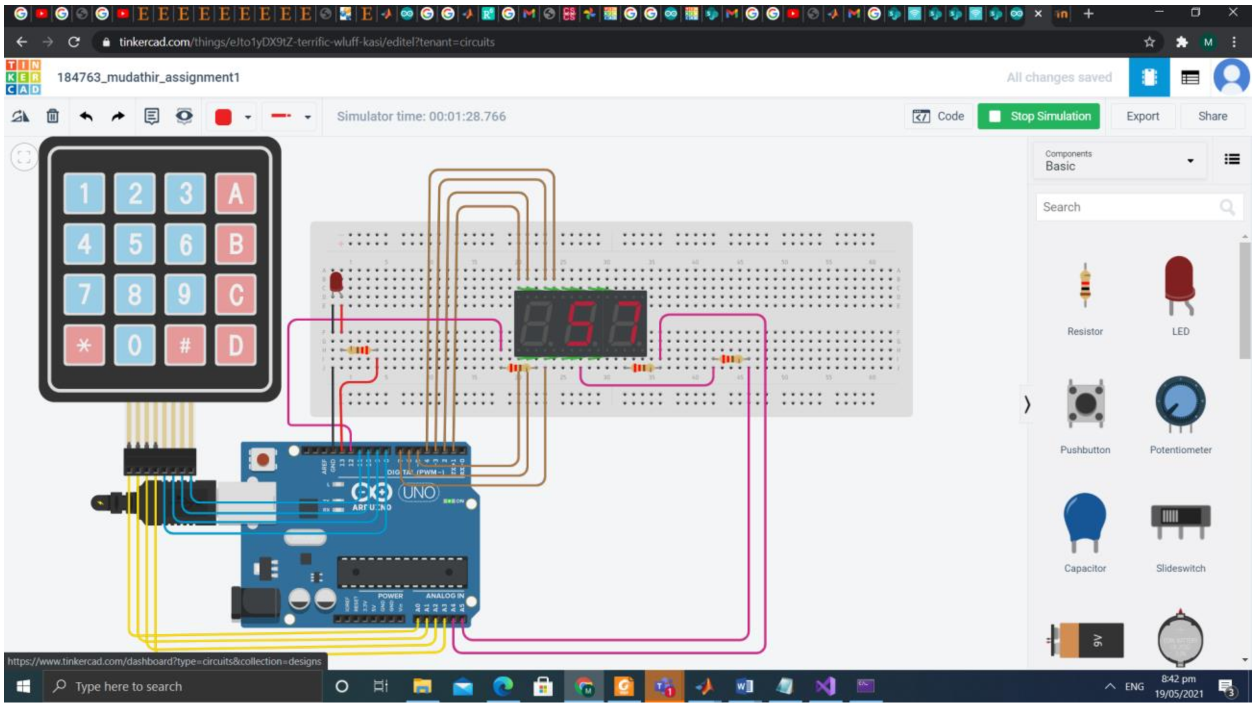
ENTERS 47



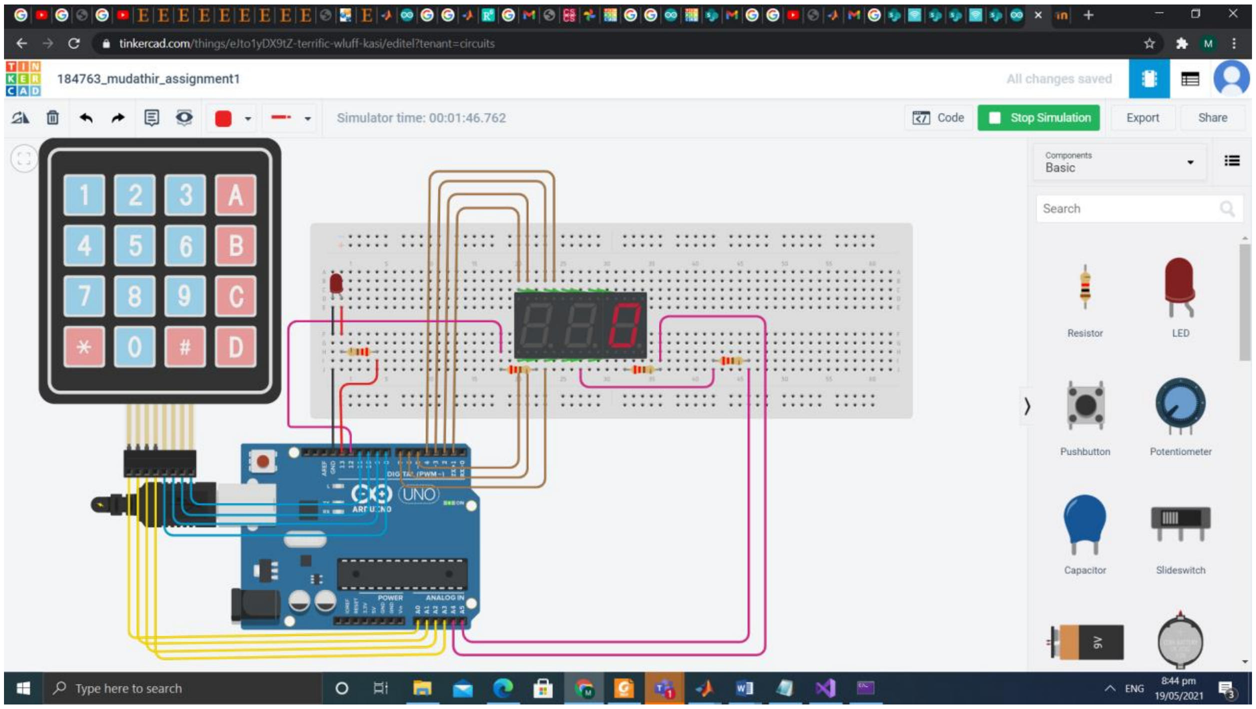
PRESSES A then C



ENTERS 25, PRESSES B AND THEN PRESSES C


tinkercad.com/things/eJto1yDX9tZ-terrific-wluff-kasi/editel?tenant=circuits
184763_mudathir_assignment1
All changes saved
Simulator time: 00:01:28.766
Code
Stop Simulation
Export
Share
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
https://www.tinkercad.com/dashboard?type=circuits&collection=designs
Type here to search
8:42 pm
19/05/2021

PRESSES D AND THE PRESSES C



BILLS OF MATERIALS

Materials-----------------------quantity

ATMega 328 microcontroller -   1

Seven segment display – 3

Keypad-1

LED-1

220 ohm resistors -4

Breadboard -1

Connecting wires – 20

CODE

```
//initialize buttons
const char buttons[4][4]={{'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
{'*','0','#','D'}};
uint8_t prevstates[4][4]={};
const uint8_t datamatrix[10]={0b11111100,0b10010000,0b01111010,
0b11011010,0b10010110,0b11001110,0b11101110,0b10011000,0b11111110,
0b11011110};
//control pins for the seven segment
const int control[3]={A4,A5,12};
const uint8_t overflow=13;//overflow pin
int value=0;//variable to sum up entered digits from keypad
uint8_t count=0;//acts like a signal to keep the digits displays on
uint8_t sum_signal=0;//acts like a signal to display current sum
int current_sum=0;
//signal to allow users keep entering numbers after pressing key 'C'
uint8_t input_mode=0;
int prev_value =0;//stores previous value to sum/subtract from current sum
void setup()
{
  DDRB|=0b00001111;//sets pins 8-11 as output
  PORTB|=0b00001111;//sets pins 8-11 HIGH
  DDRC&=0b11110000;//sets pins A0-A3 as input
  PORTC|=0b00001111;//sets pins A0-A3 as INPUT_PULLUP
  DDRD|=0b11111110;//sets pins 1-7 as output
  for(int i=0;i<3;i++)//sets control pins to output
    pinMode(control[i],OUTPUT);
  for(int i=0;i<3;i++)//sets control pins HIGH(off mode)
    digitalWrite(control[i],HIGH);
  pinMode(overflow,OUTPUT);// sets overflow pin as output

}
void loop()
{
  for(int col=0;col<4;col++)
```

```
{
 digitalWrite(col+8,LOW);
 for(int row=0;row<4;row++)
 {
  uint8_t state=!(PINC&(1<<row));
  if(state &&!prevstates[row][col])
  {
   //checks if input mode is activated to enter numbers after displaying current sum
   if(buttons[row][col]>='0'&& buttons[row][col]<='9'&& input_mode==1){
    value=0;//sets value to 0 to input new numbers
    count=1;//activates normal display
    input_mode=0;//deactivates input mode until c is pressed again
    sum_signal=0;//deactivates current_sum display
   }
   if(buttons[row][col]>='0'&& buttons[row][col]<='9'){
   //sums up entered digits
   value=(value*10)+(buttons[row][col]-'0');
   if(value>=0 && value <= 999)//checks if there is no overflow
      count=1;//activates normal display signal
   else{//if overflow occurs
    count=0;//deactivates normal  display signal
    blink(overflow,4);//blinks overflow led 4 times
    value=0;//sets value to 0 to input new numbers

   }
   }
   if(buttons[row][col]=='A')//addition operation
   {
    count=0;//deactivates normal display signal
    }

      prev_value=value;//stores previous number

      current_sum+=prev_value;//adds  number to sum

      value=0;//sets value to 0 to input new numbers

    }

    if(buttons[row][col]=='B')//subtraction operation

    {

      count=0;//deactivates normal display signal

      prev_value=value;//stores previous number

      current_sum-=prev_value;//subtracts number from sum

      value=0;//sets value to 0 to input new numbers

    }

    if(buttons[row][col]=='C')//current sum operation

    {

      count=0;//deactivates normal display signal

      sum_signal=1;//activates sum display

    }

    if(buttons[row][col]=='D')//reset operation

    {

      count=0;//deactivates normal display signal

      current_sum=0;// puts current sum to zero

      value=0;//sets value to 0 to input new numbers

      sum_signal=0;//deactivates sum signal

      input_mode=0;//deactivates input mode

    }




  }

  prevstates[row][col]=state;
```

```
    }
    digitalWrite(col+8,HIGH);


  }
  if(count)//checks if display signal is activated
    display(value);//calls display function
  if(sum_signal){//checks if sum_signal is activated
    if(current_sum<0){//checks if sum is negative
      display(0);//displays 0
      current_sum=0;//puts current sum to zero
    }
    else if(current_sum>999){//checks for current sum overflow
      blink(overflow,4);//blinks overflow LED
      current_sum=0;//puts current sum to zero
    }
    else
    display(current_sum);//displays current sum if no overlow or negatives
    input_mode=1;//activates input mode to allow new numbers to be entered
  }
}
void display(int c)//function to display on seven segment using multiplexing
{
          int j=0;
          if(c==0)//checks if number is 0
          {
            digitalWrite(control[j],LOW);//seven segment on
            PORTD&=0;//clears display
            PORTD|=datamatrix[c];//writes number to display
            delay(1);
          digitalWrite(control[j],HIGH);//seven segment off
         }

         for(;c!=0;)//displays numbers greater than zero
         {
          digitalWrite(control[j],LOW);//seven segment on
          PORTD&=0;//clears display
          PORTD|=datamatrix[c%10];//digit extraction
          delay(1);
          digitalWrite(control[j],HIGH);//seven segment off
          c=c/10;//for 2nd and 3rd digit extraction
          j++;//transfers control to the next seven segment

         }
}
void blink(int led,int n)//blink function
{
  for(int i=0;i<n;i++){
  digitalWrite(led,HIGH);
  delay(500);
  digitalWrite(led,LOW);
  delay(500);
  }
}
```