```
/*
Color Sensor     Arduino
-----------      --------
 VCC              5V
 GND               GND
 s0             VCC
 s1             VCC
 s2             12  1
 s3             11  2
 OUT              10  3
 OE              GND

 Servo 180       Arduino
 ---------       -------
 VCC              5v
 GND               GND
 Sig1              6
 Sig2             46

Keypad         Arduino
------         -------
r1             A0
r2             A1
r3             A2
r4             A3
```

```
c1          A7
c2          A6
c3          A5
c4          A4

LCD        Arduino
---       ---------
D[0]~D[3]    53~50
D[4]~D[7]     10~13
Enable      20
R/W        38
Bipolar(RS)    19
Anod        VCC
Cathod       Ground
VSS        Ground
V0         Ground
VDD        VCC

*/
//modse variables
int mode=0;
int redmax,yellowmax,greenmax;



//Second Motor angle Values
```

```
#define Mred  40000-3400
#define Mgreen 40000-2800
#define Myellow 40000-2350
#define Mundef 40000-1900


//LCD and Keypad Defines
#define MrLCDsCrib PORTB
#define DataDir_MrLCDsCrib DDRB
#define MrLCDsControl PORTD
#define DataDir_MrLCDsControl DDRD
#define LightSwitch 1
#define ReadWrite 7
#define BiPolarMood 2

int r1=A0;
int r2=A1;
int r3=A2;
int r4=A3;

int c1=A7;
int c2=A6;
int c3=A5;
int c4=A4;
int cnt=0;
```

```
char arr[4];
char pass[4]={'1','2','3','4'};
char buffers1[50];
char buffers2[50];
char buffers3[50];

//Sensor Variables
//Constants
const int s2 = 1;
const int s3 = 2;
const int out = 3;


// Variables
int red = 0;
int green = 0;
int blue = 0;
long first;
long second;
int init_mod;
int r0,g0,b0;
int RedC=0;
int GreenC=0;
int YellowC=0;
```

```
void setup() {
  pinMode(22,OUTPUT);
  //Motor 1 Initialization

  DDRH = 0b00001000;
  TCCR4A=TCCR4B=TCCR4C=0;
  TCCR4A |= ( 1 << COM4A1) |(1 <<COM4A0) |(1<<WGM41);
  TCCR4B |= (1<<WGM42) |(1<<WGM43) |(1<<CS41);
  ICR4 = 39999;
  OCR4A= 40000-2800;  //first pos


  //Motor 2 Initialization
  DDRL = 0b00001000;
  TCCR5A=TCCR5B=TCCR5C=0;
  TCCR5A |= ( 1 << COM5A1) |(1 <<COM5A0) |(1<<WGM51);
  TCCR5B |= (1<<WGM52) |(1<<WGM53) |(1<<CS51);
  ICR5 = 39999;
  OCR5A= Mred;  //first pos
  //Sensor initiaization

  Serial.begin(9600);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  //Lcd AND Keypad initialization
  pinMode(r1,OUTPUT);
  pinMode(r2,OUTPUT);
  pinMode(r3,OUTPUT);
  pinMode(r4,OUTPUT);

  pinMode(c1,INPUT);
  pinMode(c2,INPUT);
  pinMode(c3,INPUT);
  pinMode(c4,INPUT);

  initial_LCD();

}


void loop()
{
 while(1)
  {
    getPassword();
    if(cnt==4)
    {
     if(verifyPassword()==true)
```

```
   {
     Send_A_Command(0x01); //Clear Screen 0x01 = 00000001
     delay(2);

     Send_A_String("Correct Password ",100);
     delay(2000);

     Send_A_Command(0x01);
     delay(2);

////////////////mode choose///////////////////////////////
     Send_A_String("Choose mod(1~4)",100);
     Send_A_Command(0xc0 | 0);
     char *q=new char[1];
     do{
       q[0]=getpressed();
      }
     while(q[0]!='1'&&q[0]!='2'&&q[0]!='3'&&q[0]!='4');
     Send_A_String(q,2);
     mode=q[0]-'0';
///////////////////////////////////////////

     Send_A_Command(0x01); //Clear Screen 0x01 = 00000001
     delay(2);
```

```
    break;
  }
  else
  {
    Send_A_Command(0x01); //Clear Screen 0x01 = 00000001
    delay(2);
    wrong();
    Send_A_String("Wrong Password ",50);
    delay(2000);
    Send_A_Command(0x01); //Clear Screen 0x01 = 00000001
    delay(2);
    cnt=0;
    initial_LCD();
  }
 }
}

  if (mode==1)
  {
    Send_A_String("Red Green Yellow",100);
    while(1)
   {
     updatelcd();
     OCR4A = 40000-2800;
     delay(1000);
   OCR4A = 40000-3400;
   delay(750);

   int c=sensor_start(); //0 :red 1:green 2:yellow
   switch(c)
   {
    case 0:
     OCR5A= Mred;
     break;
    case 1:
     OCR5A= Mgreen;
     break;
    case 2:
     OCR5A= Myellow;
     break;
   }

   OCR4A = 40000-3900;
   delay(1000);

 }
}
else if (mode==2) //choose an order for all the cups
{
 mode2init();
```

```
Send_A_String("Red Green Yellow",100);
while(redmax-RedC>0||greenmax-GreenC>0||yellowmax-YellowC>0)
  {
  updatelcd();
  OCR4A = 40000-2800;
  delay(1000);
  OCR4A = 40000-3400;
  delay(750);

  int c=sensor_start(); //0 :red 1:green 2:yellow
  if (c==0&&RedC>redmax) c=3;
  if (c==1&&GreenC>greenmax) c=3;
  if (c==2&&YellowC>yellowmax) c=3;
  switch(c)
  {
   case 0:
    OCR5A= Mred;
    break;
   case 1:
    OCR5A= Mgreen;
    break;
   case 2:
    OCR5A= Myellow;
    break;
   case 3:
     OCR5A= Mundef;
     break;
   }

   OCR4A = 40000-3900;
   delay(1000);
 }
}
else if (mode==3)
{
 mode2init();
 Send_A_String("Red Green Yellow",100);
 int Red[3];
 int Green[3];
 int Yellow[3];
 for (int i=0;i<3;i++)
  Red[i]=Green[i]=Yellow[i]=0;

 while(!(Red[2]>=redmax&&Green[2]>=greenmax&&Yellow[2]>=yellowmax))
   {
   updatelcd();
   OCR4A = 40000-2800;
   delay(1000);
   OCR4A = 40000-3400;
   delay(750);
```

```
int c=sensor_start(); //0 :red 1:green 2:yellow

if (c==0)

{

  if (Red[0]<redmax)

  {

    Red[0]++;

    OCR5A= Mred;

  }

  else if (Red[1]<redmax)

  {

    Red[1]++;

    OCR5A= Mgreen;

  }

  else if (Red[2]<redmax)

  {

    Red[2]++;

    OCR5A= Myellow;

  }

  else

    OCR5A= Mundef;

}

else if (c==1)

{
  if (Green[0]<greenmax)
  {
    Green[0]++;
    OCR5A= Mred;
  }
  else if (Green[1]<greenmax)
  {
    Green[1]++;
    OCR5A= Mgreen;
  }
  else if (Green[2]<greenmax)
  {
    Green[2]++;
    OCR5A= Myellow;
  }
  else
    OCR5A= Mundef;
}

else if (c==2)
{

  if (Yellow[0]<yellowmax)
  {
    Yellow[0]++;
    OCR5A= Mred;
   }
   else if (Yellow[1]<yellowmax)
   {
    Yellow[1]++;
    OCR5A= Mgreen;
   }
   else if (Yellow[2]<yellowmax)
   {
    Yellow[2]++;
    OCR5A= Myellow;
   }
   else
    OCR5A= Mundef;
  }


  OCR4A = 40000-3900;
  delay(1000);
  }
 }
 else if (mode==4)
 {
   Send_A_String("Initilaization !",100);
   Send_A_Command(0xc0 | 1);
```

```
        Send_A_String("PrsAnyKeyToGo...",100);



        OCR5A = Mred;
        getpressed();
        delay(500);
        OCR5A = Mgreen;
        getpressed();
        delay(500);
        OCR5A = Myellow;
        getpressed();
        delay(500);
        OCR5A =Mundef;  // right (feweshy)
        getpressed();
        delay(500);
      }

  reset();
}

void reset()
{
 OCR5A= Mred;
 OCR4A= 40000-2800;
 mode=0;
 RedC=0;

 GreenC=0;

 YellowC=0;

 cnt=0;

 initial_LCD();

}


void mode2init()

{/*

 Send_A_Command(0x01); //Clear Screen 0x01 = 00000001

 delay(2);

 ///////////////red value choose//////////////////////////////

 delay(250);

 Send_A_String("Enter Red Value",2);

 char *r=new char[2];

 r[0]=r[1]=0;

 int i=0;

 char z='0';

 Send_A_Command(0xc0 | 0);

 do{

 z=getpressed();

 if (z-'0'>=0&&z-'0'<=9&&i<2)

 {

   Send_A_Command(0xc0 | i);

   Send_A_Character(z);
```

```
  r[i++]=z;

}

delay(500);

}while(z-'0'>=0&&z-'0'<=9&&i<2);

r[i]=0;

sscanf(r, "%d", &redmax);

Serial.println(r);

Serial.println(redmax,DEC);

//////////////////////////////////////////

Send_A_Command(0x01); //Clear Screen 0x01 = 00000001

delay(2);    */


//new code


    Send_A_Command(0x01); //Clear Screen 0x01 = 00000001

    delay(2);

  /////////////////red value choose////////////////////////////////


    Send_A_String("Enter Red Value",100);


    char x;

    do

    {

    Send_A_Command(0xc0 | 0);

    x=getpressed();
```

```
   }while(!(x-'0'>=0&&x-'9'<=0));

   Send_A_String(&x,2);

   redmax=x-'0';


   //////////////////////////////////////////

  Send_A_Command(0x01); //Clear Screen 0x01 = 00000001

  delay(2);

////////////////Green value choose//////////////////////////////


   Send_A_String("Enter Green Value",100);


   do

   {

   Send_A_Command(0xc0 | 0);

   x=getpressed();

   }while(!(x-'0'>=0&&x-'9'<=0));

   Send_A_String(&x,2);

   greenmax=x-'0';


   //////////////////////////////////////////

  Send_A_Command(0x01); //Clear Screen 0x01 = 00000001

  delay(2);

  ////////////////red value choose//////////////////////////////


   Send_A_String("Enter Yellow Value",100);
```

```
        do
        {
        Send_A_Command(0xc0 | 0);
        x=getpressed();
        }while(!(x-'0'>=0&&x-'9'<=0));
        Send_A_String(&x,2);
        yellowmax=x-'0';

        ////////////////////////////////////////
       Send_A_Command(0x01); //Clear Screen 0x01 = 00000001
       delay(2);

}

void updatelcd()
{
  Send_A_Command(0xc0 | 0);
  Send_A_String("                ",2);       //to clear only the second row
  Send_A_Command(0xc0 | 1);

  itoa(RedC, buffers1,10);
  delay(2);
  Send_A_String(buffers1,2);
  Send_A_Command(0xc0 |6);
  itoa(GreenC, buffers2,10);
  delay(2);
  Send_A_String(buffers2,2);

  Send_A_Command(0xc0 |12);
  itoa(YellowC, buffers3,10);
  delay(2);
  Send_A_String(buffers3,2);
}
int sensor_start()
{
  for (int i=0;i<36;i++)
  {
    delay(100);
    color();
    red+=r0;
    green+=g0;
    blue+=b0;
  }
  red/=36;
  green/=36;
  blue/=36;

  Serial.print("R Intensity:");    //to show in serial monitor
  Serial.print(red, DEC);

  Serial.print(" G Intensity: ");

  Serial.print(green, DEC);

  Serial.print(" B Intensity : ");

  Serial.print(blue, DEC);


  if (blue>=32&&green>=32) //29 32

    {

      RedC++;

      Serial.println(" - (Red Color)");

      return 0;

    }

  else if (blue<32&&blue>24&&green<=32) //blue 23

     {

      GreenC++;

      Serial.println(" - (Green Color)");

      return 1;

     }

  else if (blue<=24)

  {

    YellowC++;

    Serial.println(" - (Yellow Color)");

    return 2;

  }

  else
```

```
  {
    Serial.println();
    return sensor_start();
  }
}

void color()
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  //count frequency of the red color
  r0 = Pulse_In(out);
  digitalWrite(s3, HIGH);
  //count frequency of the blue color
  b0 = Pulse_In(out);
  digitalWrite(s2, HIGH);
  //count frequency of the green color
  g0 = Pulse_In(out);
}

int Pulse_In(int out)
{
 init_mod= digitalRead(out);
 while(digitalRead(out)==init_mod)
 {}
 first=micros();
 while(digitalRead(out)!=init_mod)
 {}
second=micros();
 return (second-first);
}



bool verifyPassword()
 {
  return arr[0]==pass[0] && arr[1]==pass[1] && arr[2]==pass[2] && arr[3]==pass[3];
}

void getPassword()
{
  Send_A_Command(0xc0 | cnt);
  char *x=new char[1];
    x[0]=getpressed();
   Send_A_String(x,2);
   arr[cnt]=x[0];
   cnt++;
   delay(500);
   delete[]x;
}



void initial_LCD()
{
   DataDir_MrLCDsControl |= 1<<LightSwitch | 1<<ReadWrite | 1<<BiPolarMood;
 delay(15);

  Send_A_Command(0x01); //Clear Screen 0x01 = 00000001
  delay(200);     //2
  Send_A_Command(0x38);     //// Letting micro controller to know there are 8 bits of data
  delay(20);     //50
  Send_A_Command(0b00001110);    // (1110) mn el shemal ll yemin First 1 for starting the LED,second for the display on,third for cursor on and , fourth for Blinking cursor
  delay(20);     //50
   Send_A_String("Enter Password",100);
  }

void Check_IF_MrLCD_isBusy()
{
  DataDir_MrLCDsCrib = 0;
  MrLCDsControl |= 1<<ReadWrite;
  MrLCDsControl &= ~1<<BiPolarMood;

  while (MrLCDsCrib >= 0x80)
 {
  Peek_A_Boo();
 }

 DataDir_MrLCDsCrib = 0xFF; //0xFF means 0b11111111
}



void Peek_A_Boo()
{
 MrLCDsControl |= 1<<LightSwitch;
 asm volatile ("nop");
 asm volatile ("nop");
 MrLCDsControl &= ~1<<LightSwitch;
}

void Send_A_Command(unsigned char command)
{
 Check_IF_MrLCD_isBusy();
 MrLCDsCrib = command;
 MrLCDsControl &= ~ ((1<<ReadWrite)|(1<<BiPolarMood));
 Peek_A_Boo();
 MrLCDsCrib = 0;
}
```

```
void Send_A_Character(unsigned char character)
{
  Check_IF_MrLCD_isBusy();
  MrLCDsCrib = character;
  MrLCDsControl &= ~ (1<<ReadWrite);
  MrLCDsControl  |= 1<<BiPolarMood;
  Peek_A_Boo();
  MrLCDsCrib = 0;
}


void Send_A_String(char *StringOfCharacters,int _delay)
{
  while(*StringOfCharacters > 0)
  {
    Send_A_Character(*StringOfCharacters++);
    delay(max(_delay,2));
  }
}

char getpressed()
{
  //setting the columns as high initially
  digitalWrite(c1,HIGH);
  digitalWrite(c2,HIGH);

  digitalWrite(c3,HIGH);

  digitalWrite(c4,HIGH);


  //checking everything one by one


  //case 1: col1 =0 while other col as 1

  digitalWrite(r1,LOW);

  digitalWrite(r2,HIGH);

  digitalWrite(r3,HIGH);

  digitalWrite(r4,HIGH);

  //checking each column for row1 one by one

  if(digitalRead(c1)==0)

    {

      tick();

      return'1';

    }

  else if(digitalRead(c2)==0)

    {

      tick();

      return '2';

    }

  else if(digitalRead(c3)==0)

  {

    tick();
    return '3';
  }
  else if(digitalRead(c4)==0)
  {
    tick();
    return 'A';
  }

  //case 2: col2 =0 while other col as 1
  digitalWrite(r1,HIGH);
  digitalWrite(r2,LOW);
  digitalWrite(r3,HIGH);
  digitalWrite(r4,HIGH);
  //checking each column for row1 one by one
  if(digitalRead(c1)==0)
    {
      tick();
      return '4';
    }
  else if(digitalRead(c2)==0)
    {
      tick();
      return '5';
    }
  else if(digitalRead(c3)==0)
```

```
 {
   tick();
   return '6';
 }
else if(digitalRead(c4)==0)
 {
   tick();
   return 'B';
 }

//case 3: col3 =0 while other col as 1
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,LOW);
digitalWrite(r4,HIGH);
//checking each column for row1 one by one
if(digitalRead(c1)==0)
 {
   tick();
   return '7';
 }
else if(digitalRead(c2)==0)
 {
   tick();
   return '8';
```

```
  }
else if(digitalRead(c3)==0)
  {
    tick();
    return '9';
  }
else if(digitalRead(c4)==0)
  {
    tick();
    return 'C';
  }

//case 4: col4 =0 while other col as 1
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,LOW);
//checking each column for row1 one by one
if(digitalRead(c1)==0)
  {
    tick();
    return '*';
  }
else if(digitalRead(c2)==0)
  {
   tick();
   return '0';
  }
 else if(digitalRead(c3)==0)
  {
   tick();
   return '#';
  }
 else if(digitalRead(c4)==0)
  {
   tick();
   return 'D';
  }

return getpressed();
}
void tick()
{
 digitalWrite(22,HIGH);
 delay(50);
 digitalWrite(22,LOW);
}
void wrong()
{
 digitalWrite(22,HIGH);
```

```
  delay(400);

  digitalWrite(22,LOW);

  delay(100);


  digitalWrite(22,HIGH);

  delay(400);

  digitalWrite(22,LOW);

}
```