```
// GLOBAL VARIABLES //

// #FOR SENSOR FREQ
unsigned int redfreq;
unsigned int bluefreq;
unsigned int greenfreq;
unsigned int digit;

// #COUNTERS

unsigned char bluecnt;
unsigned char redcnt;
unsigned char greencnt;


// #LCD
unsigned char bluecnt_lcd;
unsigned char redcnt_lcd;
unsigned char greencnt_lcd;
unsigned char test;

// #SENSOR
unsigned int redtest ;
unsigned int bluetest ;
unsigned int greentest ;


//////////////////////
////////FUNCTIONS//////
//////////////////////

//DELAY
void Delay_us(unsigned int x)
{
 unsigned int i;
for ( i = 0 ; i<x ; i++)
 x=x ;
}

// LCD
  // #TO PASS DATA# //
void lcd_data (unsigned char daata)
{
```

```
  PORTC = daata ;// asscii !!
  PORTD = PORTD | 0x01 ;// RS Data go to reg of lc1d --> RS=1
  PORTD = PORTD & 0xFD ;//RW Writing data --> RW=0
  PORTD = PORTD | 0x04 ;//EN =1

  Delay_us(5000); // to make sure that the char has passed
  PORTD = PORTD & 0xFB ; // EN = 0
}
 // #TO PASS INSTRUCTIONS# //
void lcd_cmd (unsigned char cmd)
{

  PORTC = cmd ;// to change the inst
  PORTD = PORTD & 0xFC ;//RS = 0 data passing goingt to , RW = 0  Write on LCD
  PORTD = PORTD | 0x04 ;//EN =1
  Delay_us(5000); // to make sure that the char has passed
  PORTD = PORTD & 0xFB ; // EN = 0
}
 // #TO INITIALZE THE LCD# //
void initial ()
{

 lcd_cmd (0x38); // to make it 2 * 16
 lcd_cmd (0x06); // increment curser
 lcd_cmd (0x0C); //LCD on & off
 lcd_cmd (0x01); //on screen
}
 // #TO STRING# //
 void lcd_string(const unsigned char *str, unsigned char num)
{

unsigned char i;

for(i=0;i<num;i++)
{
lcd_data(str[i]);
  Delay_us(50000);
}

}


// INTERUPT
```

```
  // /#HELPING FUNCTIPNS FOR INTERRUPT#/
void Delay_us_int(unsigned int x)
{
 unsigned int i;
for ( i = 0 ; i<x ; i++)
 x=x ;
}
void lcd_cmd_int (unsigned char cmd)
{

  PORTC = cmd ;// to change the inst
  PORTD = PORTD & 0xFC ;//RS = 0 data passing goingt to , RW = 0  Write on LCD
  PORTD = PORTD | 0x04 ;//EN =1
  Delay_us_int(5000); // to make sure that the char has passed
  PORTD = PORTD & 0xFB ; // EN = 0
}

void lcd_data_int (unsigned char daata)
{

  PORTC = daata ;// asscii !!
  PORTD = PORTD | 0x01 ;// RS Data go to reg of lc1d --> RS=1
  PORTD = PORTD & 0xFD ;//RW Writing data --> RW=0
  PORTD = PORTD | 0x04 ;//EN =1

  Delay_us_int(5000); // to make sure that the char has passed
  PORTD = PORTD & 0xFB ; // EN = 0
}

void initial_int ()
{

 lcd_cmd_int (0x38); // to make it 2 * 16
 lcd_cmd_int (0x06); // increment curser
 lcd_cmd_int (0x0C); //LCD on & off
 lcd_cmd_int (0x01); //on screen
}

 void lcd_string_int(const unsigned char *str, unsigned char num)
{

unsigned char i;

for(i=0;i<num;i++)
```

```
{
lcd_data_int(str[i]);
  Delay_us_int(55000);
}

}

  // /THE MAIN INTERRUPT FUNCTION/
void interrupt()
 {
 unsigned int i;

for ( i = 0 ; i<5000 ; i++)
  i=i ;


 lcd_cmd_int (0x01); // CLEAR LCD
 bluecnt_lcd = bluecnt + 48;
 redcnt_lcd = redcnt + 48;
 greencnt_lcd = greencnt + 48;


  lcd_cmd_int (0x83); // first row third col
  lcd_string_int("Blue=",5);
  lcd_data_int(bluecnt_lcd);


 lcd_cmd_int (0x8B);
 lcd_string_int("red=",4);
 lcd_data_int(redcnt_lcd);


 lcd_cmd_int (0xC4);//??     should move forward!
 lcd_string_int("green=",6);
 lcd_cmd_int (0xCA);
 lcd_data_int(greencnt_lcd);
 lcd_string_int("    ",4);

 lcd_cmd_int (0x01); // CLEAR LCD

 for ( i = 0 ; i<5000 ; i++)
  i=i ;
 INTCON = INTCON & 0XFD; //CLEAR INTF
```

```
 }

//SERVO MOTORS

// /#SERVO1#/
void Rotation0()
{
unsigned int i;
for(i=0;i<50;i++)
{
PORTB = PORTB | 0x02;   // 0000 0010
Delay_us(1050); // pulse of 800us (DUTY CYCLE )
PORTB = PORTB & 0xFD;   // 1111 1101
Delay_us(18950);
}


}

void Rotation90()
{
unsigned int i;
for(i=0;i<50;i++)
{
PORTB = PORTB | 0x02;
Delay_us(1800); // pulse of 1500us
PORTB = PORTB & 0xFD;
Delay_us(18200);
}
}

void Rotation180()

{
unsigned int i;
for(i=0;i<50;i++)
{
PORTB = PORTB | 0x02;
Delay_us(2200); // pulse of 2200us
PORTB = PORTB & 0xFD;
Delay_us(17800);
```

```
}
}

// /#SERVO2#/

 void Rotation2_0()
{
unsigned int i;
for(i=0;i<50;i++)
{
PORTB = PORTB | 0x04;   // 0000 0100
Delay_us(1200); // pulse of 800us (DUTY CYCLE )
PORTB = PORTB & 0xFB;   // 1111 1011
Delay_us(18800);
}


}

void Rotation2_90()
{
unsigned int i;
for(i=0;i<50;i++)
{
PORTB = PORTB | 0x04;
Delay_us(1650); // pulse of 1500us
PORTB = PORTB & 0xFB;
Delay_us(18350);
}
}

 void Rotation2_180()

{
unsigned int i;
for(i=0;i<50;i++)
{
PORTB = PORTB | 0x04;
Delay_us(2200); // pulse of 2200us
PORTB = PORTB & 0xFB;
Delay_us(17800);
}
}
```

```
//SENSOR
// /#BLUE#/
void pick_blue()


{

    PORTA = PORTA & 0XF3; //1111 0011
    bluefreq = 0 ;
    while((PORTB & 0x10));      // out rising  0001 0000

    while(!(PORTB & 0x10));      // out falling

    while((PORTB & 0x10));      // out rising

    while(!(PORTB & 0x10))
    {
     //out turns to high pulse       0001 0000
     bluefreq = bluefreq + 1;
    }




}

// /#RED#/
void pick_red()
{
    PORTA= PORTA & 0XFB;    //1111 1011
    PORTA = PORTA |0X08 ;   // 0000 1000
    redfreq = 0;
    while((PORTB & 0x10));      // out rising
    while(!(PORTB & 0x10));      // out falling
    while((PORTB & 0x10));      // out rising
     while(!(PORTB & 0x10))
    {   //out turns to high pulse
     redfreq = redfreq + 1;
     }



}
```

```
// /#GREEN#/
void pick_green()
{


    PORTA = PORTA | 0X08; //0000 1000
    PORTA = PORTA |0x04  ;//0000 0100
    greenfreq = 0;
    while((PORTB & 0x10));      // out rising
    while(!(PORTB & 0x10));      // out falling
    while((PORTB & 0x10));      // out rising
     while(!(PORTB & 0x10))
    {   //out turns to high pulse
     greenfreq = greenfreq + 1;
    }


}

//////////////////////////
//////////MAIN/////////

void main() {
 //TRISB = TRISB | 0x08; //0000 1000

// INITIAL

// #SWITCH# --> RB5

 TRISB = TRISB |0x20; //0010 0000

// #SERVO

 TRISB = TRISB & 0XFD; // PORTB RB1 1111 1101 (SERVO1)
 TRISB = TRISB & 0XFB; // PORTB RB2 1111 1011(SERVO2)

// #LCD
 TRISC = 0x00;// RC0-RC-7 for data
 PORTC = 0X00;
 TRISD = 0x00;//for rs,rw,e  such that RS=RD0 / RW=RD1 / E=RD2
 initial ();

// #SENSOR
TRISA = TRISA & 0XF0; // --> RA0-RA3 for S[0:3] outputs
```

```
TRISB = TRISB | 0X10; // --> RB4 out  0001 0000

//  #CHOOSE FREQ OF SENSOR
PORTA = PORTA | 0X01;   // s0 HIGH
PORTA = PORTA & 0XFD;   // S1 LOW

// #LED

TRISB = TRISB &  0xF7 ; // first led 1 -> RB3
TRISD = TRISD & 0xAF ; // second led 3 -> RD4  ,third led 4-> RD6


// STRATING POSITIONS
 Rotation2_0()  ;
 Rotation180()  ;


// WELCOMING
 lcd_cmd (0x84); // first row third col
 lcd_string("WELCOME",7);
 lcd_cmd (0xC4);
 lcd_string("DR.BELAL",8);

     // LED OFF
     PORTB = PORTB & 0xF7;
     PORTD = PORTD  & 0xEF;
     PORTD = PORTD & 0xBF;
     Delay_us(500000);

      //INITIALIZE THE COUNTERS OF EACH COLOR!
      bluecnt=0;
      redcnt=0;
      greencnt=0;


while(1)

 {

 // CHECK IF OFF
 if((PORTB & 0x20))
 {
   bluecnt=0;
   redcnt=0;
```

```
  greencnt = 0 ;
  continue;
 }

 // CLEAR LCD
 lcd_cmd (0x01);

 //SECOND POSITION FOR SERVO1
Rotation90();
Delay_us(500000);
     // INITAL VALUES FOR FREQ
      redfreq=0;
      bluefreq=0;
      greenfreq= 0;

      //INT ENABLE
      INTCON =INTCON | 0X90; //SET GIE AND INTE


      // READ RED FREQ
       pick_red();
       Delay_us(500000);
      //READ GREEN FREQ
       pick_green();
       Delay_us(500000);
      //READ BLUE FREQ
       pick_blue();
       Delay_us(500000);


      // IF RED
      if(redfreq > bluefreq && redfreq > greenfreq)
      {
         // SELECT RED BASKET
         Rotation2_180();
         // RED LED ON
        PORTB = PORTB  | 0x08; // 0000 1000
        PORTD = PORTD  & 0xEF; // 1110 1111
        PORTD = PORTD & 0xBF; // 1011 1111
         Delay_us(5000000);
         // PRINT RED ON LCD
         lcd_cmd (0x83); // first row third col
         lcd_string("RED",3);
         // INCREMENT RED COUNTER
```

```
        redcnt++;


      }
      // IF BLUE
      else if( bluefreq > redfreq && bluefreq > greenfreq  )
      {
       // SELECT BLUE BASKET
       Rotation2_90();
       // BLUE LED ON
       PORTB = PORTB & 0xF7; // 1111 0111
       PORTD = PORTD  & 0xEF; // 1110 1111
       PORTD = PORTD | 0x40; // 0100 0000
       Delay_us(5000000);
        // PRINT BLUE ON LCD
        lcd_cmd (0x83);
        lcd_string("BLUE",4);
        // INCREMENT BLUE COUNTER
        bluecnt++;


      }

        //IF GREEN
      else if(greenfreq >=bluefreq && greenfreq >=redfreq )
       {
          // select the green bascket
          Rotation2_0();
         // green LED
         PORTB = PORTB  & 0xF7; // 1111 0111
         PORTD = PORTD  | 0x10; // 0001 0000
         PORTD = PORTD & 0xBF; // 1011 1111
          Delay_us(5000000);
         // print GREEN
         lcd_cmd (0x83); // first row third col
         lcd_string("GREEN",5);
         // INCREMENT GREEN COUNTER
         greencnt++;
      }


// THIRD POSITION FOR SERVO1
Rotation0();
Delay_us(5000000);
```

```
 // SERVO1 GO BACK TO ITS ORIGIN
 Rotation90();
 Delay_us(1100);
 Rotation180();
 Delay_us(5000000);


}


}
```