```
    // variables //
unsigned char bt_read = 0;    //  variable to save blutooth data
int sensor_read[3] ;      //  array to save data from sensors
int y , i;                //  variables to save data from sensors
int count=1 ;             //  variable to count in 7segment
const delay_sensor = 400 ;
    // functions //
void Init(void);        //  function to intialize all ports
void Forward(void);
void Back(void);
void Right(void);
void Left(void);
void move_automatic(void);
void Stop(void);
void Rotate_CW(void);         //  move gripper clockwise
void Rotate_ACW(void);        //  move gripper anti-clockwise
void gripper_up(void);
void gripper_down(void);
void gripper_open(void);
void gripper_close(void);
void seg_7(void);
void buzzer(void);

void main()
{
   Init();



   while(1)
```

```
{
   if ( (UART1_Data_Ready() == 0) && ( bt_read== 0 )) move_automatic ;
   else if (UART1_Data_Ready() == 1)
{
        bt_read = UART1_Read();
switch (bt_read)
   {
        case 'S' :
        Stop();
        break ;

        case 'A' :
        move_automatic();
        break ;

        case 'F' :
        Forward();
        break ;

        case 'B' :
        Back();
        break ;

        case 'R' :
        Right();
        break ;

        case 'L' :
        Left();
```

```
break ;

case 'U' :
gripper_up();
break ;

case 'D' :
gripper_down();
break ;

case 'O' :
gripper_open();
break ;

case 'C' :
gripper_close();
break ;

case 'Y' :
Rotate_CW();
break ;

case 'T' :
Rotate_ACW();
break ;

case 'P' :
buzzer();
break ;
```

```
            case 'N' :
            seg_7();
            break ;
        }

    }

  }
}
```

```
 void Init (void)
{
    TRISB = 0b00000000;
    PORTB = 0b00000100;       //one relay is mirrored in PCB

    TRISC = 0b00000000;
    PORTC = 0b00000000;

    TRISD = 0b00000000;
    PORTD = 0b00000000;

    UART1_Init(9600);
    ADC_Init() ;
}
```

```
void Stop(void)
{
      PORTB = 0b00000100;  // one  relay is mirrored in the PCB
      PORTC = 0 ;
      PORTD.B4 = 0 ;
}

void move_automatic(void)
{
  for (  i = 0 ; i < 3 ; i++ )
    { sensor_read[i] = ADC_Read(i); }
       for ( i = 0 ; i < 3 ; i++ )
        {
          if ( sensor_read[i] < 600 )
              sensor_read[i] = 1 ;
          else
              sensor_read[i] = 0 ;
        }
     y = sensor_read[0]*4 + sensor_read[1]*2 + sensor_read[2] ;

      switch (y)
      {
           case 2 :
           Forward();
           break ;

           case 1 :
           case 3 :
           Right();
```

```
            break ;

            case 4 :
            case 6 :
            Left();
            break;

            default :
            while (y== 0)
            {
              Back();
              delay_ms(800);
              Left();
              delay_ms(1000);
              Right();
              delay_ms(1000);
            }
            break ;
        }
            delay_ms(delay_sensor);
}
void Forward(void)
{   PORTB = 0b00001101;    }
void Back(void)
{   PORTB = 0b00000010;    }
void Left(void)
{   PORTB = 0b00001110;    }
void Right(void)
{   PORTB = 0b00000001;    }
```

```
void Gripper_up(void)
{     PORTB = 0b00010100;    }

void Gripper_down(void)
{     PORTB = 0b00100100;    }

void Gripper_close(void)
{     PORTB = 0b01000100;    }

void Gripper_open(void)
{     PORTB = 0b10000100;    }



void Rotate_CW(void)
{
   for(i=0 ; i<50 ; i++)                    //frequancy loop
   {
      PORTC = 0x02;
      Delay_us(800);
      PORTC = 0x00;
      Delay_us(19200);
   }
}

void Rotate_ACW(void)
{
   for(i=0 ; i<50 ; i++)
   {
      PORTC = 0x02;
```

```
      Delay_us(1800);
      PORTC = 0x00;
      Delay_us(19200);
   }
}
void seg_7(void)
 {
    if(count==1)
        PORTD = 0b00000001;
    else if(count==2)
        PORTD = 0b00000010;
    else if(count==3)
        PORTD = 0b00000011;
    else if(count==4)
        {
            PORTD = 0b00000100;
            count=0;
        }
  Delay_ms(500)    ;
  count++;

 }
void buzzer(void)
{
    PORTD.B4 = 1;
}
```