

```
#include "Wire.h"

#include "TPA81.h"

#include "ssd1306.h"

#include "DHT.h"

// Create new TPA81 instance

TPA81 tpa;

#define VCCSTATE SSD1306_SWITCHCAPVCC

#define WIDTH    128

#define HEIGHT   64

#define PAGES     8

#define OLED_RST  9

#define OLED_DC   8

#define OLED_CS   10

#define SPI_MOSI  11 /* connect to the DIN pin of OLED */

#define SPI_SCK   13

uint8_t oled_buf[WIDTH * HEIGHT / 8];

int gesture[8][20];

int tgesture[8];

int sum=0;

int sflag; // standby flag

int gcount;

int c;

int ex;

int vol=5;

const uint8_t v1 [] PROGMEM = {
```

[illegible]

[illegible]

[illegible]

```
const uint8_t v2 [] PROGMEM = {
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
const uint8_t v6 [] PROGMEM = {
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

switch(input)
{ case 0 : SSD1306_char3216(26, 25, '0', oled_buf); break;
case 1 : SSD1306_char3216(26, 25, '1', oled_buf); break;
case 2 : SSD1306_char3216(26, 25, '2', oled_buf); break;
case 3 : SSD1306_char3216(26, 25, '3', oled_buf); break;
case 4 : SSD1306_char3216(26, 25, '4', oled_buf); break;
case 5 : SSD1306_char3216(26, 25, '5', oled_buf); break;
case 6 : SSD1306_char3216(26, 25, '6', oled_buf); break;
case 7 : SSD1306_char3216(26, 25, '7', oled_buf); break;
case 8 : SSD1306_char3216(26, 25, '8', oled_buf); break;
case 9 : SSD1306_char3216(26, 25, '9', oled_buf); break; } }
else
{ n1=input/10;
n2=input%10;
switch(n1) {
case 0 : SSD1306_char3216(10, 25, '0', oled_buf); break;
case 1 : SSD1306_char3216(10, 25, '1', oled_buf); break;
case 2 : SSD1306_char3216(10, 25, '2', oled_buf); break;
case 3 : SSD1306_char3216(10, 25, '3', oled_buf); break;
case 4 : SSD1306_char3216(10, 25, '4', oled_buf); break;
case 5 : SSD1306_char3216(10, 25, '5', oled_buf); break;
case 6 : SSD1306_char3216(10, 25, '6', oled_buf); break;
case 7 : SSD1306_char3216(10, 25, '7', oled_buf); break;
case 8 : SSD1306_char3216(10, 25, '8', oled_buf); break;
case 9 : SSD1306_char3216(10, 25, '9', oled_buf); break; }
switch(n2) {
case 0 : SSD1306_char3216(26, 25, '0', oled_buf); break;
case 1 : SSD1306_char3216(26, 25, '1', oled_buf); break;
case 2 : SSD1306_char3216(26, 25, '2', oled_buf); break;

```

```

    case 3 : SSD1306_char3216(26, 25, '3', oled_buf); break;
    case 4 : SSD1306_char3216(26, 25, '4', oled_buf); break;
    case 5 : SSD1306_char3216(26, 25, '5', oled_buf); break;
    case 6 : SSD1306_char3216(26, 25, '6', oled_buf); break;
    case 7 : SSD1306_char3216(26, 25, '7', oled_buf); break;
    case 8 : SSD1306_char3216(26, 25, '8', oled_buf); break;
    case 9 : SSD1306_char3216(26, 25, '9', oled_buf); break; }
  }}

void setup() {
  Serial.begin(9600);
  Wire.begin();
  SSD1306_begin();
  SSD1306_clear(oled_buf);
}

void loop() {
  for (int i = 0; i <= 7; i++) // reading 8 pixel { if (tpa.getPoint(i)>29)
  {Serial.print("#");
  Serial.print(" ");
  gesture[i][0]=1;}
  else
  {Serial.print(".");
  Serial.print(" ");
  gesture[i][0]=0;}
  }

  for (int y = 0; y<=7; y++){sum+=gesture[y][0]; Serial.println(sum); // sum of 8 pixel
  Serial.println(" ");delay(100); // speed of reading

  ////////////////////////////////////////////standby

  if (sum==8){sflag=1; Serial.println("standby");}

  else {sum=0;}

```

////////////////////////////////////progress

```
if (sflag==1 && sum!=8){
  Serial.println("start gesture");
  gcount=0;
  c=0;
  sum=0;
  while (sum<8)
  {sum=0;
    for (int i = 0; i <= 7; i++) { if (tpa.getPoint(i)>29)
      {Serial.print("#");
        Serial.print(" ");
        gesture[i][gcount]=1;}
      else
      {Serial.print(".");
        Serial.print(" ");
        gesture[i][gcount]=0;}
    }
    for (int y = 0; y<=7; y++) {sum+=gesture[y][gcount];delay(5);} // sum of 8 pixel
    Serial.println(" "); delay(50); // speed of reading
    gcount++; // lentgh of gesture
  }
  Serial.print("sum : "); Serial.println(sum);
  Serial.print("lentgh of gesture : "); Serial.println(gcount);
  delay(5);
  for (int c=0; c<=7; c++){
    for (int z=0; z<=gcount; z++)
      {tgesture[c]+=gesture[c][z];delay(5);}
    Serial.println("processing");}
  tgesture[0]+= tgesture[1];
```

```

tgesture[2]+= tgesture[3];
tgesture[4]+= tgesture[5];
tgesture[6]+= tgesture[7];
tgesture[0]+= tgesture[2];
tgesture[4]+= tgesture[6];
if (tgesture[0]<tgesture[4])
    { vol+=(gcount/2);  Serial.print("volume up");}
if (tgesture[4]<tgesture[0])
    {vol=(gcount/2);  Serial.print("volume down");}
sflag=0;
}
sum=0;
SSD1306_clear(oled_buf);
switch (vol)
{ case 1 : SSD1306_bitmap(0, 0,v1, 128, 64, oled_buf); break;
  case 2 : SSD1306_bitmap(0, 0,v2, 128, 64, oled_buf); break;
  case 3 : SSD1306_bitmap(0, 0,v3, 128, 64, oled_buf); break;
  case 4 : SSD1306_bitmap(0, 0,v4, 128, 64, oled_buf); break;
  case 5 : SSD1306_bitmap(0, 0,v5, 128, 64, oled_buf); break;
  case 6 : SSD1306_bitmap(0, 0,v6, 128, 64, oled_buf); break;
  case 7 : SSD1306_bitmap(0, 0,v7, 128, 64, oled_buf); break;
  case 8 : SSD1306_bitmap(0, 0,v8, 128, 64, oled_buf); break;
  case 9 : SSD1306_bitmap(0, 0,v9, 128, 64, oled_buf); break;
  case 10 : SSD1306_bitmap(0, 0,v10, 128, 64, oled_buf); break;
  case 11 : SSD1306_bitmap(0, 0,v11, 128, 64, oled_buf); break; }
SSD1306_display(oled_buf);
}

```