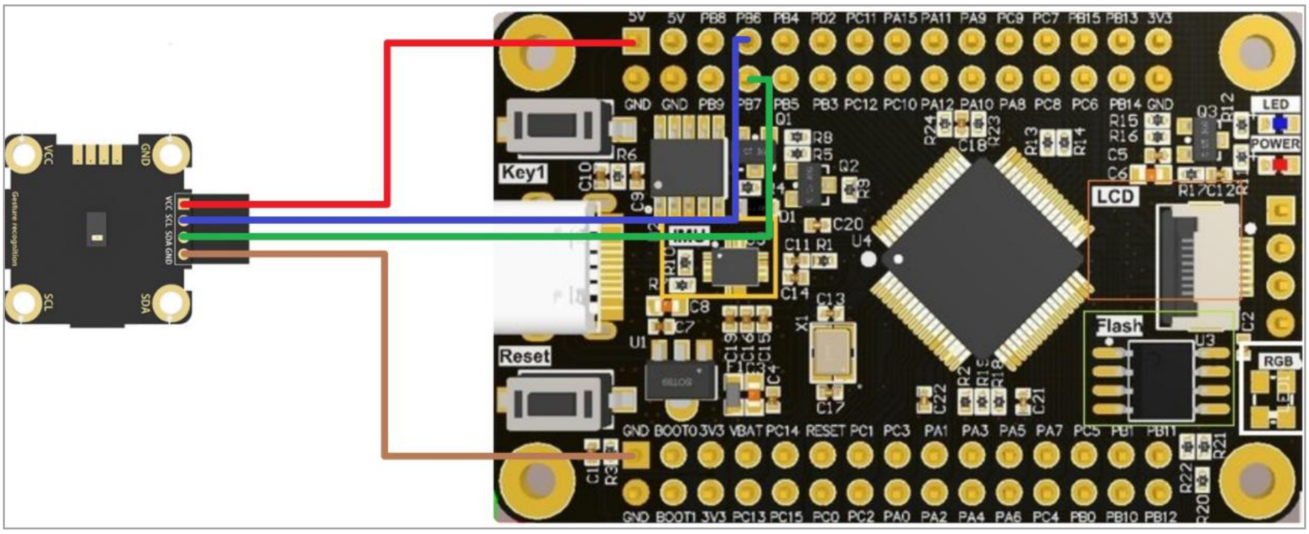
Gesture recognition

Learning objectives

In this course, we mainly learn to implement gesture recognition functions using STM32F103RCT6 and gesture recognition module modules.

Prepare before class



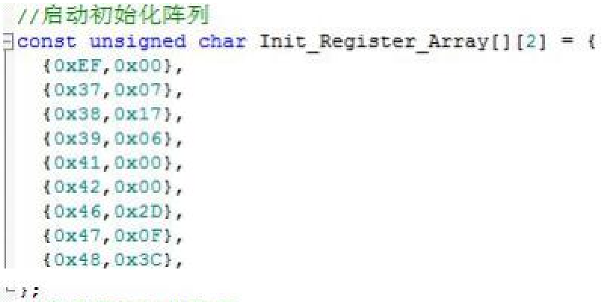
The gesture recognition module adopts I2C communication, and the SDA and SCL of the module are connected to the PB7 and PB6 pins of the STM32F103RCT6 board.

Program

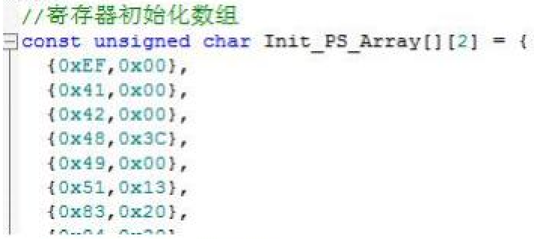
The I2C address of the module is 0x73, and the last bit needs to be reserved as a read and write bit, so it is shifted to the left.

```
#define I2C_ADDR                    0x73 << 1   //语音识别模块地址,模块地址为0x73,
```

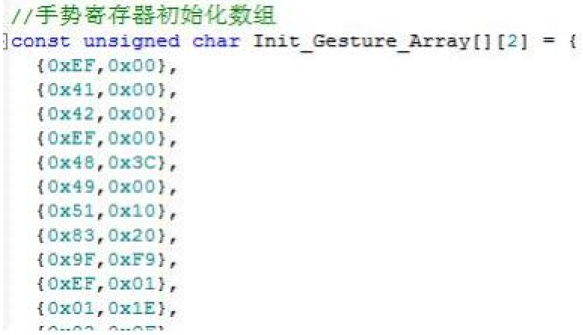
Define initialization arrays, register arrays, gesture register addresses.



```
//启动初始化阵列
const unsigned char Init_Register_Array[][2] = {
    {0xEF,0x00},
    {0x37,0x07},
    {0x38,0x17},
    {0x39,0x06},
    {0x41,0x00},
    {0x42,0x00},
    {0x46,0x2D},
    {0x47,0x0F},
    {0x48,0x3C},

};
```

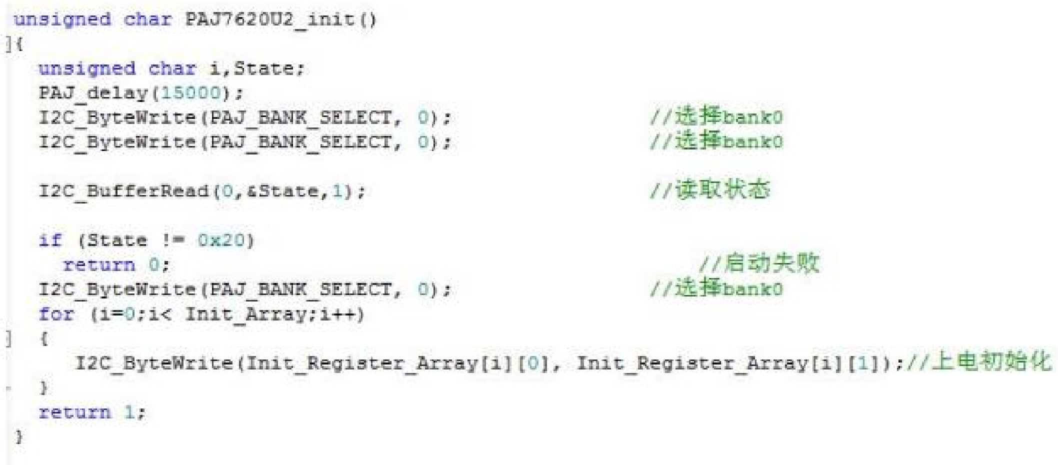


```
//寄存器初始化数组
const unsigned char Init_PS_Array[][2] = {
    {0xEF,0x00},
    {0x41,0x00},
    {0x42,0x00},
    {0x48,0x3C},
    {0x49,0x00},
    {0x51,0x13},
    {0x83,0x20},
```



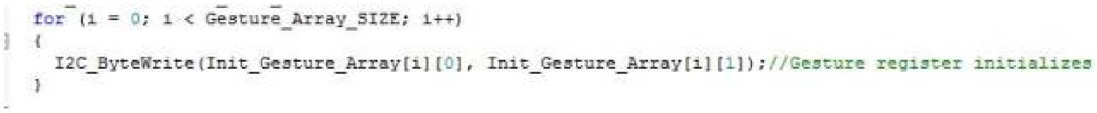
```
//手势寄存器初始化数组
const unsigned char Init_Gesture_Array[][2] = {
    {0xEF,0x00},
    {0x41,0x00},
    {0x42,0x00},
    {0xEF,0x00},
    {0x48,0x3C},
    {0x49,0x00},
    {0x51,0x10},
    {0x83,0x20},
    {0x9F,0xF9},
    {0xEF,0x01},
    {0x01,0x1E},
```

Initialize the gesture recognition module, return 1 for success and 0 for failure.



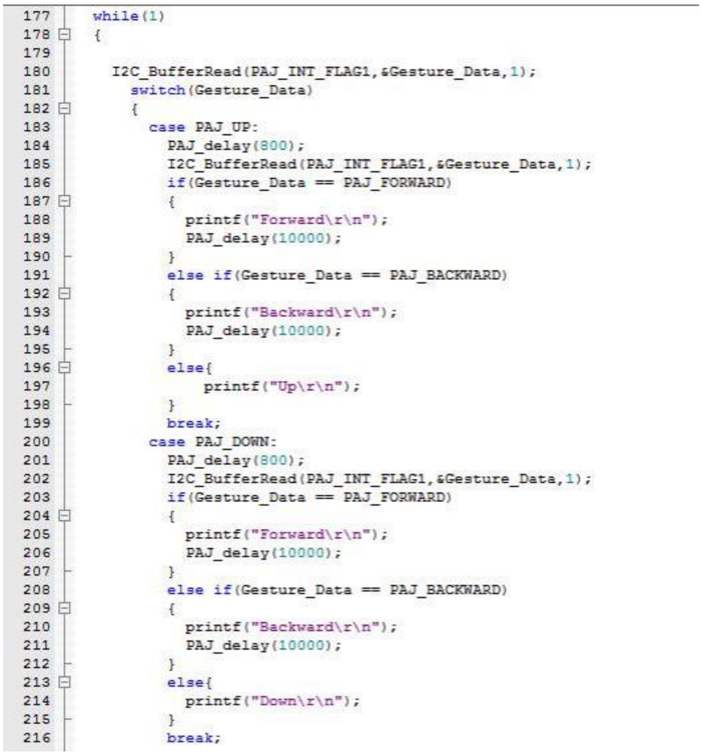
```
unsigned char PAJ7620U2_init()
{
    unsigned char i,State;
    PAJ_delay(15000);
    I2C_ByteWrite(PAJ_BANK_SELECT, 0);                //选择bank0
    I2C_ByteWrite(PAJ_BANK_SELECT, 0);                //选择bank0

    I2C_BufferRead(0,&State,1);                       //读取状态

    if (State != 0x20)
      return 0;                                           //启动失败
    I2C_ByteWrite(PAJ_BANK_SELECT, 0);                //选择bank0
    for (i=0;i< Init_Array;i++)
    {
       I2C_ByteWrite(Init_Register_Array[i][0], Init_Register_Array[i][1]);//上电初始化
    }
    return 1;
}
```

Gesture recognition register value writes.

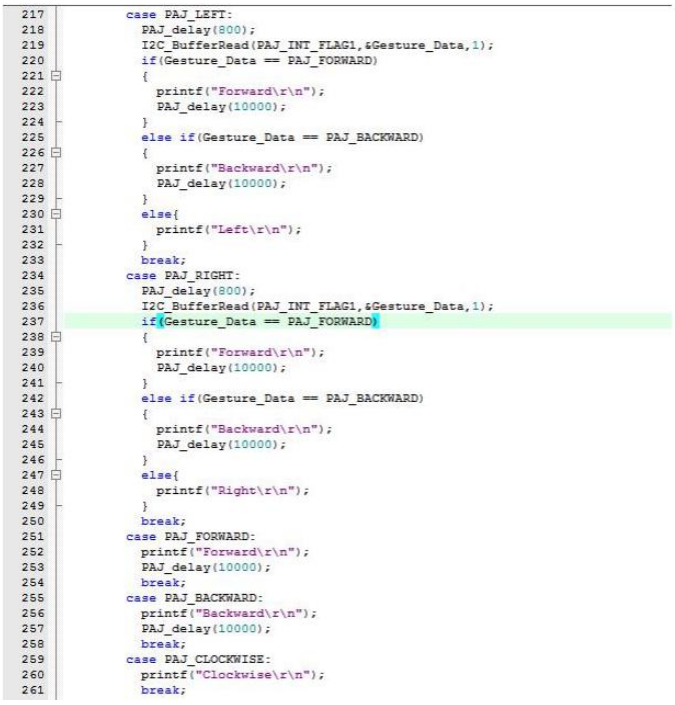


```
  for (i = 0; i < Gesture_Array_SIZE; i++)
  {
    I2C_ByteWrite(Init_Gesture_Array[i][0], Init_Gesture_Array[i][1]);//Gesture register initializes
  }
```

By reading the value recognized by the gesture, and then judging it through the switch statement, and then printing out the corresponding gesture name by the serial port.



```
while(1)
{

  I2C_BufferRead(PAJ_INT_FLAG1,&Gesture_Data,1);
    switch(Gesture_Data)
    {
      case PAJ_UP:
        PAJ_delay(800);
        I2C_BufferRead(PAJ_INT_FLAG1,&Gesture_Data,1);
        if(Gesture_Data == PAJ_FORWARD)
        {
          printf("Forward\r\n");
          PAJ_delay(10000);
        }
        else if(Gesture_Data == PAJ_BACKWARD)
        {
          printf("Backward\r\n");
          PAJ_delay(10000);
        }
        else{
            printf("Up\r\n");
        }
        break;
      case PAJ_DOWN:
        PAJ_delay(800);
        I2C_BufferRead(PAJ_INT_FLAG1,&Gesture_Data,1);
        if(Gesture_Data == PAJ_FORWARD)
        {
          printf("Forward\r\n");
          PAJ_delay(10000);
        }
        else if(Gesture_Data == PAJ_BACKWARD)
        {
          printf("Backward\r\n");
          PAJ_delay(10000);
        }
        else{
          printf("Down\r\n");
        }
        break;
```



```
      case PAJ_LEFT:
        PAJ_delay(800);
        I2C_BufferRead(PAJ_INT_FLAG1,&Gesture_Data,1);
        if(Gesture_Data == PAJ_FORWARD)
        {
          printf("Forward\r\n");
          PAJ_delay(10000);
        }
        else if(Gesture_Data == PAJ_BACKWARD)
        {
          printf("Backward\r\n");
          PAJ_delay(10000);
        }
        else{
          printf("Left\r\n");
        }
        break;
      case PAJ_RIGHT:
        PAJ_delay(800);
        I2C_BufferRead(PAJ_INT_FLAG1,&Gesture_Data,1);
        if(Gesture_Data == PAJ_FORWARD)
        {
          printf("Forward\r\n");
          PAJ_delay(10000);
        }
        else if(Gesture_Data == PAJ_BACKWARD)
        {
          printf("Backward\r\n");
          PAJ_delay(10000);
        }
        else{
          printf("Right\r\n");
        }
        break;
      case PAJ_FORWARD:
        printf("Forward\r\n");
        PAJ_delay(10000);
        break;
      case PAJ_BACKWARD:
        printf("Backward\r\n");
        PAJ_delay(10000);
        break;
      case PAJ_CLOCKWISE:
        printf("Clockwise\r\n");
        break;
```

Experimental phenomenon

After the program is downloaded, it will run it, and if the module is successfully initialized, it will print "Gesture Sensor OK", otherwise "Gesture Sensor Error" will be printed. If initialization fails, you can reset the program under STM32 to reinitialize. After successful initialization, the value of gesture recognition is judged, and different gestures will print out different action names through the serial port. Position the gesture recognition module upright, spread your palm to face the module, and swipe from left to right from the front of the module to print "Left". Swipe right to left from the front of the module to print "Right". Swipe from the front of the module from bottom to top to print "Up". Swipe down from the front of the module from top to bottom, and "Down" is printed. Approaching from the front of the module from back to front, "Forward" is printed. From the front of the module from front to back, "Backward" is printed. Clench your fist and point two or three fingers directly in front of the module, then make a short clockwise turn to print "Clockwise." Clench your fist and point two or three fingers directly in front of the module, then make a short counterclockwise turn to print "AntiClockwise." Wave your hand directly in front of the module for a while, and print "Wave".