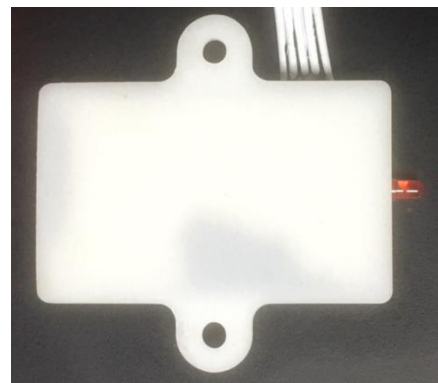
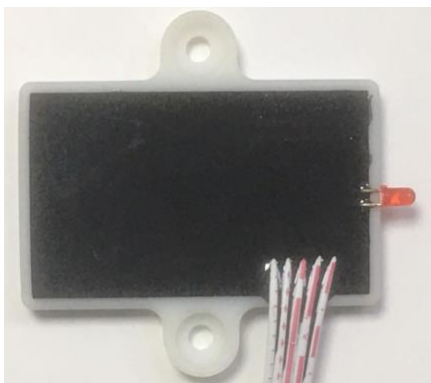




下雨感知模塊
RAIN_MODULE
(專利產品)



特性

模塊特性

- 工作電壓：3V~5V
- 待機電流：小於1mA
- 工作環境：-10~85度C
- 儲存環境：-40~100度C
- 可做無水狀態標定
- 可設定感應靈敏度

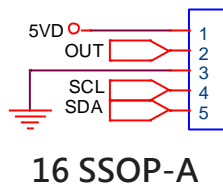
周邊特性

- 具有雨水感測專用IO輸出
- 具有I2C通信介面可做設定與標定
- 可以多種接線應用 4Pin 或 5Pin

概述

該模塊主要應用在遮雨棚、電動窗簾、電動門窗、汽車天窗、等室外需遮雨貨避雨設備系統。利用PCB上的銅箔自身產生的電容值，與外界導電物質接觸發生感應，導致電容值變化，針對銅箔自身電容值變化，利用專用晶片量測資料做相關的應用；如果導電物質恰好是水的時候，進而可利用這個原理製作一個實用的雨滴感應模組。

引腳圖



引腳說明

引腳順序	引腳名稱	功能	屬性	說明
1	5VD	電源	PWR	模塊電源正
2	OUT	下雨感知輸出	O	
3	GND	電源	PWR	模塊電源負
4	SCL	I2C面	O	
5	SDA	I2C面	I/O	

極限參數

電源供應電壓.....	VSS-2.8V~5.5V
埠輸入電壓.....	VSS-0.3V~VDD+0.3V
儲存溫度.....	-50°C~125°C
工作溫度.....	-20°C~85°C

DOUT參數

當檢測到有下雨時DOUT 輸出低電平，否則為高電平

I2C命令參數

讀取模塊狀態

通過 IIC 輸出數據，共發送 2 筆數據，第一筆為水位數據（無水時輸出 00H，有水時輸出 01H），第二筆是校驗和（水位數據+1）

標定模塊乾燥(無水)狀態

重新無水校正有效指令：01H 00H 00H CHECKSUM

設定模塊水感靈敏度(值越小越靈敏)

修改檢測靈敏值：02H THRESHOLD_L 00H CHECKSUM

第一筆數據必須為 02H

第二筆數據是靈敏度值低八位，有效值為 5-100，超過範圍不進行改寫

第三筆數據，必須為 0

上位機程序範例(8051)

```

unsigned int xdata data0,data1;
unsigned char xdata WAIT_SLAVE_ACK_F,SET_SEN;
/***** PT7C4663 I2C start *****/
void IIC_START()
{KEY_SDA=1; KEY_SCL=1;
  KEY_SDA=0; delay(2);
  KEY_SCL=0; delay(2);}

/***** PT7C4663 I2C stop *****/
void IIC_STOP()
{KEY_SDA=0; KEY_SCL=0;
  KEY_SCL=1; delay(2);
  KEY_SDA=1; delay(2);}

/***** WAIT_SLAVE_ACK 等待從機應答 *****/
unsigned char WAIT_SLAVE_ACK(void)
{KEY_SCL=0; KEY_SDA=1; delay(2);
  KEY_SCL=1;
  if(KEY_SDA==1){WAIT_SLAVE_ACK_F=1;} //從機無應答
  else {WAIT_SLAVE_ACK_F=0;} //從機應答
  delay(2);
  while(!KEY_SCL);
  KEY_SCL=0;
  delay(2);
  return WAIT_SLAVE_ACK_F;}

```

```

/***** MASTER_SEND_ACK 主機發送應答 *****/
void MASTER_SEND_ACK(void)
{KEY_SCL=0; KEY_SDA=1; delay(2);
  KEY_SDA=0; delay(2);
  KEY_SCL=1; delay(2);
  KEY_SCL=0; KEY_SDA=1; delay(2); delay(2);}

/***** MASTER_SEND_NOACK 主機發送無應答 *****/
void MASTER_SEND_NOACK(void)
{KEY_SDA=0; KEY_SCL=0; delay(2);
  KEY_SDA=1; delay(2);
  KEY_SCL=1; delay(2);
  KEY_SCL=0; delay(2);
  KEY_SDA=0; delay(2);
  delay(2);}

/***** IIC_WRITE_BYTE 寫一字節 *****/
void IIC_WRITE_BYTE(unsigned char IIC_WRITE_DATA)
{unsigned char i;
  for(i=0; i<8; i++)
    {if(IIC_WRITE_DATA&0x80){KEY_SDA=1;}
     else {KEY_SDA=0;}
      delay(5);
      KEY_SCL=1; delay(2);
      while(!KEY_SCL);
      KEY_SCL=0;
      IIC_WRITE_DATA<<=1;}}

/***** IIC_READ_BYTE 讀一字節 *****/
unsigned char IIC_READ_BYTE(void)
{unsigned char i=8;
  unsigned char IIC_READ_DATA=0;
  for(i=0; i<8; i++)
    {KEY_SCL=1; delay(5);
      while(!KEY_SCL);
      IIC_READ_DATA <<= 1;
      if(KEY_SDA){IIC_READ_DATA|=0x01;}
      KEY_SCL=0; delay(2);}
  return IIC_READ_DATA;}

/***** I2C MODULE READ *****/
void KEY_I2C_Read(void)
{IIC_START();
  IIC_WRITE_BYTE(0xa0+1); //寫從機地址
  WAIT_SLAVE_ACK(); //等待從機應答
  if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
  data0=IIC_READ_BYTE(); MASTER_SEND_ACK(); //讀數據到寄存器
  data1=IIC_READ_BYTE(); MASTER_SEND_NOACK(); //讀數據到寄存器
  IIC_STOP();}

```

```

/***** MODULE_I2C_CAL_ZERO *****/
void MODULE_I2C_CAL_ZERO(void)
{IIC_START();
 IIC_WRITE_BYTE(0xa0);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x01);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x00);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x00);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x02); // =0x01+0x00+0x00+1
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_STOP();}

```

```

/***** MODULE_I2C_SET_SEN *****/
void MODULE_I2C_SET_SEN(void)
{IIC_START();
 IIC_WRITE_BYTE(0xa0);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x02);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x0a);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x00);
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_WRITE_BYTE(0x0d); // =0x02+0x0a+0x00+1
 WAIT_SLAVE_ACK(); //等待從機應答
 if(WAIT_SLAVE_ACK_F){IIC_STOP(); return;}
 IIC_STOP();}

```