YC31xx SSC 应用说明

V1.0

Yichip Microelectronics

Revision History

Version	Date	Author	Description
V1.0	2020-3-2	Duanziyang	Initial version

Confidentiality Level:

confidential

目录

1. 文档说明 4
1.1 编写目的 4
1.2 适用范围 4
1.3 文件说明 4
2. 结构体说明 5
2.1 TAMPER_InitTypeDef 5
3. 库函数说明 6
3.1 SSC_Interval 6
3.2 SSC_LPMSheildingAlarmEnable 6
3.3 SSC_ClearKeyCMD 6
3.4 SSC_TemperInit 7
3.5 SSC_LPMTemperCmd 7
3.6 SSC_SensorDur 7
3.7 SSC_SensorDelay 8
3.8 SSC_LPMSensorCmd 8
3.9 SSC_LPMKeyRead 9
3.10 SSC_LPMKeyWrite 9
3.11 SSC_LPMLock 9
3.12 SSC_GetLPMStatusReg 10
3.13 SSC_LPMClearStatusBit 10
3.14 SSC_SecureCmd 11
3.15 SSC_GetSecureStatus 11
4. Demo 函数说明 11
1.4.1 ssc_enable 11
1.4.2 tamper_test 12
1.4.3 sensor_test 12
1.4.4 BPK_RW_test 13
1.4.5 SEC_IRQHandler 13

1. 文档说明

1.1 编写目的

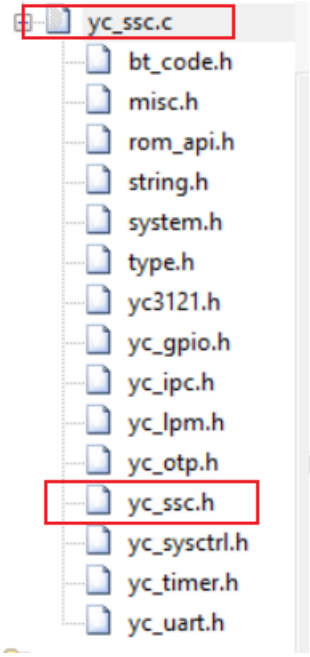
为使用 SSC 相关 Demo 及 SSC 库函数 提供指南

1.2 适用范围

31xx 系列芯片

1.3 文件说明

SSC Demo 路径为 ModuleDemo\BPU\SENSOR_TEST
SSC 库文件为如下图 ssc.c 与 ssc.h,路径为 Librarier\sdk



2. 结构体说明

2.1 TAMPER_InitTypeDef

元素名称	类型	说明	参数项
TAMPER_Port_mode	uint32_t	tamper 模式选项 选中右方参数即为选择动态模式，否则为静止模式	**1.**TAMPER_Port_None_active **2.**TAMPER_Port_S01_active **3.**TAMPER_Port_S23_active **4.**TAMPER_Port_S45_active **5.**TAMPER_Port_S67_active **6.**TAMPER_Port_ALL_active
TAMPER_Port_PullUp	uint32_t	tamper 上拉选项，若 tamper 选为静止模式，则该 tamper 必须上拉	**1.**TAMPER_Port_S0_PU **2.**TAMPER_Port_S1_PU **3.**TAMPER_Port_S2_PU **4.**TAMPER_Port_S3_PU **5.**TAMPER_Port_S4_PU **6.**TAMPER_Port_S5_PU **7.**TAMPER_Port_S6_PU **8.**TAMPER_Port_S7_PU **9.**TAMPER_Port_All_PU
TAMPER_Port_Enable	uint32_t	Tamper 开关选项	**1.**TAMPER_Port_S01 **2.**TAMPER_Port_S23 **3.**TAMPER_Port_S45 **4.**TAMPER_Port_S67 **5.**TAMPER_Port_ALL
TAMPER_GlitchTimes	uint32_t	警报持续时间窗口	**1.**TAMPER_GlitchTimes_31_25US **2.**TAMPER_GlitchTimes_1ms **3.**TAMPER_GlitchTimes_4ms **4.**TAMPER_GlitchTimes_8ms
TAMPER_PUPU_HoldTime	uint32_t	检测时间窗口，控制每个 interval, SDIO 的拉高时间	**1.**TAMPER_PUPU_HoldTime_always **2.**TAMPER_PUPU_HoldTime_2ms **3.**TAMPER_PUPU_HoldTime_8ms **4.**TAMPER_PUPU_HoldTime_16ms

3. 库函数说明

3.1 SSC_Interval

函数原型：void SSC_Interval(uint32_t Interval);
说明： 设置 ssc 自动检测周期

参数	方向	说明
uint32_t Interval	IN	SSC 检测周期，可取参数列表如下： **1.**INTERVAL_999MS **2.**INTERVAL_500MS **3.**INTERVAL_250MS **4.**INTERVAL_125MS

表格 3-1-1 SSC_Interval 形参表

返回值	说明
None	None

表格 3-1-2 SSC_Interval 返回值

3.2 SSC_LPMSheildingAlarmEnable

函数原型：void SSC_LPMSheildingAlarmEnable(FunctionalState NewState);
说明： tamper 自动检测开关

参数	方向	说明
FunctionalState NewState	IN	开关选项，可选参数列表如下： **1.**ENABLE **2.**DISABLE

表格 3-2-1 SSC_LPMSheildingAlarmEnable 形参表

返回值	说明
None	None

表格 3-2-2 SSC_LPMSheildingAlarmEnable 返回值

3.3 SSC_ClearKeyCMD

函数原型：void SSC_ClearKeyCMD(FunctionalState NewState);
说明： 硬件警报自动擦除 KEY 功能开关

参数	方向	说明
FunctionalState NewState	IN	开关选项，可选参数列表如下： **1.**ENABLE **2.**DISABLE

表格 3-3-1 SSC_ClearKeyCMD 形参表

返回值	说明
None	None

表格 3-3-2 SSC_ClearKeyCMD 返回值

3.4 SSC_TemperInit

函数原型：void SSC_TemperInit(TAMPER_InitTypeDef * TAMPER_InitStruct);
说明： 初始化 Temper（外部 sensor）

参数	方向	说明
TAMPER_InitTypeDef* TAMPER_InitStruct	IN	该指针指向外部已经配置好的 TAMPER_InitTypeDef 结构体地址

表格 3-4-1 SSC_TemperInit 形参表

返回值	说明
None	None

表格 3-4-1 SSC_TemperInit 返回值

3.5 SSC_LPMTemperCmd

函数原型：void SSC_LPMTemperCmd(uint32_t SENSOR_Port, FunctionalState NewState);
说明： Temper 开关（外部 sensor）

参数	方向	说明
uint32_t SENSOR_Port	IN	Temper 端口，可选以下参数： **1.** TAMPER_Port_S01 **2.** TAMPER_Port_S23 **3.** TAMPER_Port_S45 **4.** TAMPER_Port_S67 **5.** TAMPER_Port_ALL
FunctionalState NewState	IN	开关选项，可选参数列表如下： **1.**ENABLE **2.**DISABLE

表格 3-5-1 SSC_LPMTemperCmd 形参表

返回值	说明
None	None

表格 3-5-2 SSC_LPMTemperCmd 返回值

3.6 SSC_SensorDur

函数原型：void SSC_SensorDur(uint32_t sensor_dur);
说明： 设置 Sensor 检测时间

参数	方向	说明
uint32_t sensor_dur	IN	Sensor 检测时间，可选以下参数： **1.**SENSOR_DUR_ALWAYS_ON **2.**SENSOR_DUR_2MS **3.**SENSOR_DUR_8MS **4.**SENSOR_DUR_16MS

表格 3-6-1 SSC_SensorDur 形参表

返回值	说明
None	None

表格 3-6-2 SSC_SensorDur 返回值

3.7 SSC_SensorDelay

函数原型：void SSC_SensorDelay(uint32_t sensor_dur);
说明： 设置连续警报时间阈值，超过该时间触发警报

参数	方向	说明
uint32_t sensor_dur	IN	Sensor 连续警报时间阈值，可选以下参数： **1.** SENEOR_DELAY_31_25US **2.** SENEOR_DELAY_250US **3.** SENEOR_DELAY_1_MS **4.** SENEOR_DELAY_4MS

表格 3-7-1 SSC_SensorDelay 形参表

返回值	说明
None	None

表格 3-7-2 SSC_SensorDelay 返回值

3.8 SSC_LPMSensorCmd

函数原型：void SSC_LPMSensorCmd(uint32_t LPM_secsure_sensor,
FunctionalState NewState);
说明： LPM sensor 开关

参数	方向	说明
uint32_t LPM_secsure_sensor	IN	LPM sensor，可选选项如下： **1.**LPM_BAT_VDT12L_ENABLE **2.**LPM_BAT_VDT33H_ENABLE **3.**LPM_BAT_VDT33L_ENABLE **4.**LPM_TEMPERATURE_40_ENABLE **5.**LPM_TEMPERATURE_120_ENABLE **6.**LPM_SENSOR_ALL_ENABLE
FunctionalState NewState	IN	开关选项，可选参数列表如下： **1.**ENABLE **2.**DISABLE

表格 3-8-1 SSC_LPMSensorCmd 形参表

返回值	说明
None	None

表格 3-8-2 SSC_LPMSensorCmd 返回值

3.9 SSC_LPMKeyRead

函数原型：void SSC_LPMKeyRead(uint32_t * buf, uint8_t len);
说明： 从电池域寄存器读取密钥

参数	方向	说明
uint32_t * buf	OUT	密钥存放地址
uint8_t len	IN	读取密钥长度

表格 3-9-1 SSC_LPMKeyRead 形参表

返回值	说明
None	None

表格 3-9-2 SSC_LPMKeyRead 返回值

3.10 SSC_LPMKeyWrite

函数原型：void SSC_LPMKeyWrite(uint32_t * buf, uint32_t len);
说明： 将密钥写入电池域寄存器

参数	方向	说明
uint32_t * buf	IN	待写入密钥存放地址
uint32_t len	IN	待写入密钥长度

表格 3-10-1 SSC_LPMKeyWrite 形参表

返回值	说明
None	None

表格 3-10-2 SSC_LPMKeyWrite 返回值

3.11 SSC_LPMLock

函数原型：void SSC_LPMLock(void);
说明： 锁定 LPM sensor\tamper&key 寄存器配置，无法解锁(锁定后只有纽扣电池重新上电才能修改安全相关寄存器的配置)

参数	方向	说明
None	IN	None

表格 3-11-1 SSC_LPMLock 形参表

返回值	说明
None	None

表格 3-11-2 SSC_LPMLock 返回值

3.12 SSC_GetLPMStatusReg

函数原型：int16_t SSC_GetLPMStatusReg(void);
说明： 获取 ssc lpm sensor&tamper 中断状态

SSC_GetLPMStatusReg **返回值定义**

```
#define SSC_IT_VDT12L                ((uint16_t)0x0001)
#define SSC_IT_VDT33H                ((uint16_t)0x0002)
#define SSC_IT_VDT33L                ((uint16_t)0x0004)
#define SSC_IT_TEMPERATURE_120       ((uint16_t)0x0008)
#define SSC_IT_TEMPERATURE_40        ((uint16_t)0x0010)
#define SSC_IT_TAMPER_S0             ((uint16_t)0x0020)
#define SSC_IT_TAMPER_S1             ((uint16_t)0x0040)
#define SSC_IT_TAMPER_S2             ((uint16_t)0x0080)
#define SSC_IT_TAMPER_S3             ((uint16_t)0x0100)
#define SSC_IT_TAMPER_S4             ((uint16_t)0x0200)
#define SSC_IT_TAMPER_S5             ((uint16_t)0x0400)
#define SSC_IT_TAMPER_S6             ((uint16_t)0x0800)
#define SSC_IT_TAMPER_S7             ((uint16_t)0x1000)
#define SSC_IT_MESH_SHIELDING        ((uint16_t)0x2000)//mesh shileding alarm
```

3.13 SSC_LPMClearStatusBit

函数原型：void SSC_LPMClearStatusBit(void);
说明： 清除 lpm_intr。清除 lpm_intr 中断之前，应先清除 lpm_intr 的下级中断，包括 rtc_intr, sensor_alert,shield_alarm ，否则会继续进入中断

3.14 SSC_SecureCmd

函数原型：void SSC_SecureCmd(uint32_t SSC_secsure, FunctionalState NewState);
说明： SSC secure sensor 开关

参数	方向	说明
uint32_t SSC_secsure	IN	SSC sensor，可选以下参数： **1.**SSC_SECSURE__VDT12L_ENABLE **2.**SSC_SECSURE__VDT33H_ENABLE **3.**SSC_SECSURE__VDT33L_ENABLE **4.**SSC_BAT__VDT12L_ENABLE **5.**SSC_BAT__VDT33H_ENABLE **6.**SSC_BAT__VDT33L_ENABLE **7.**SSC_BAT__TEMPERATURE_120_ENABLE **8.**SSC_BAT__TEMPERATURE_40_ENABLE **9.**SSC_SECURE_SENSOR_ALL_ENABLE
FunctionalState NewState	IN	开关选项，可选参数列表如下： **1.**ENABLE **2.**DISABLE

表格 3-14-1 SSC_SecureCmd 形参表

返回值	说明
None	None

表格 3-14-2 SSC_SecureCmd 返回值

3.15 SSC_GetSecureStatus

函数原型：uint8_t SSC_GetSecureStatus(void);
说明： 获取 SSC secure sensor 状态

SSC_GetSecureStatus 返回值定义

```
#define SSC_IT_LPM_VDT12L                ((uint8_t)0x01)
#define SSC_IT_SECSURE_VDT33H            ((uint8_t)0x02)
#define SSC_IT_SECSURE_VDT33L            ((uint8_t)0x04)
#define SSC_IT_BAT_VDT12L                ((uint8_t)0x08)
#define SSC_IT_BAT_VDT33H                ((uint8_t)0x10)
#define SSC_IT_BAT_VDT33L                ((uint8_t)0x20)
#define SSC_IT_BAT_TEMPERATURE_120       ((uint8_t)0x40)
#define SSC_IT_BAT_TEMPERATURE_40        ((uint8_t)0x80)
```

4. Demo 函数说明

1.4.1 ssc_enable

该函数设置 sensor 硬件自动检测周期并使能检测
void ssc_enable(void)

```
{
    SSC_Interval(INTERVAL_999MS);
    SSC_LPMSheildingAlarmEnable(ENABLE);
}
```

1.4.2 tamper_test

该函数设置 Tamper 0，1，2，3 为动态模式，4，5，6，7 为静态模式（必须上拉）
并设置警报持续时间窗口为 4ms（警报时间超过改时间触发报警），检测窗口为 8ms。

```
void tamper_test(void)
{
    TAMPER_InitTypeDef TAMPER_InitStruct;

    TAMPER_InitStruct.TAMPER_Port_mode =
            TAMPER_Port_S01_active | TAMPER_Port_S23_active;
    TAMPER_InitStruct.TAMPER_Port_PullUp =
            TAMPER_Port_S4_PU | TAMPER_Port_S5_PU |\
            TAMPER_Port_S6_PU | TAMPER_Port_S7_PU;
    TAMPER_InitStruct.TAMPER_Port_Enable     = TAMPER_Port_ALL;
    TAMPER_InitStruct.TAMPER_GlitchTimes     = TAMPER_GlitchTimes_4ms;
    TAMPER_InitStruct.TAMPER_PUPU_HoldTime = TAMPER_PUPU_HoldTime_8ms;

    SSC_TemperInit(&TAMPER_InitStruct);
}
```

1.4.3 sensor_test

该函数设置 sensor 检测时间为 8MS，报警持续时间为 4MS，并打开所有传感器开关

```
void sensor_test(void)
{
    SSC_SensorDur(SENSOR_DUR_8MS);
    SSC_SensorDelay(SENEOR_DELAY_4MS);
    SSC_LPMSensorCmd(LPM_BAT_VDT12L_ENABLE | LPM_BAT_VDT33H_ENABLE | \
                     LPM_BAT_VDT33L_ENABLE    |    LPM_TEMPERATURE_40_ENABLE    |    \
  LPM_TEMPERATURE_120_ENABLE, ENABLE);
}
```

1.4.4 BPK_RW_test

该函数测试密钥写入及读取

```
void BPK_RW_test(void)
{
    MyPrintf("f8414 LPM_GPIO_WKHI is %08x\n", lpm_read(LPM_GPIO_WKHI));
    SSC_LPMKeyWrite(Datain, 32);
    SSC_LPMKeyRead(Dataout, 32);
    printv(Dataout, 32, "Dataout");
    MyPrintf("f8414 LPM_GPIO_WKHI is %08x\n", lpm_read(LPM_GPIO_WKHI));
}
```

1.4.5 SEC_IRQHandler

中断服务函数，中断触发时，读取密钥，并清除lpm_intr

```
void SEC_IRQHandler(void)
{
    MyPrintf("SEC_IRQHandler In\n");
    SSC_LPMKeyRead(Dataout, 32);
    printv(Dataout, 32, "Dataout");
    SSC_LPMClearStatusBit();
}
```