



觅感科技-AT581x 参考代码使用说明 V5.0

本文档主要介绍如何觅感科技 AT581x 雷达模块 IIC 接口参数配置的方法和说明。

目录

1、概述.....	3
2、 代码移植和使用	4
2.1、代码移植	4
2.2、代码使用	5
3、 雷达参数调试说明	7
3.1、软件复位	7
3.2、工作功耗	7
3.3、感应时间	8
3.4、保护时间	8
3.5、增益	8
3.6、感应阈值	9
3.7、光敏	9
3.8、检测窗口	10
3.9、RF 开关	10
3.10、频点配置	10
3.11、自激	10
3.12、探底	11
3.13、余量	12
4、 版本信息	13

1、概述

AT581x 包含以下两个型号：AT5812 和 AT5815。AT581x 使用 IIC 进行配置，觅感科技的参考代码对 AT581x 常用功能进行了封装，用户只需针对自己使用的 MCU 对延时函数和 IIC 的 IO 进行配置，即可快速实现对 AT581x 的使用。

MoreSense

2、代码移植和使用

2.1、代码移植

1、适配延时函数 void DelayUs(unsigned char Time);

```
time.c
1  #include "time.h"
2
3
4
5  /*-----*/
6  * 函数名称: DelayUs
7  * 功能: 短延时函数
8  * 输入参数: Time延时时间长度 延时时长Time*2Us
9  * 返回参数: 无
10 /*-----*/
11 void DelayUs(unsigned char Time)
12 {
13     unsigned char a;
14     for(a=0;a<Time;a++)
15     {
16         NOP();
17     }
18 }
19
20 /*-----*/
```

2、适配 IIC

```
IIC.H
1  #ifndef __IIC_H__
2  #define __IIC_H__
3
4  #include "time.h"
5
6
7  #define IIC_SCL      RA1  // 定义SCL引脚
8  #define IIC_SDA      RA0  // 定义SDA引脚
9
10 #define SDA_OUT      TRISA0 =0  // 配置SDA为输出
11 #define SDA_IN        TRISA0 =1; WPUA0= 1  // 配置SDA为输入上拉
12
13 #define SCL_OUT      TRISA1 =0  // 配置SCL为输出
14 #define SCL_IN        TRISA1 =1; WPUA1= 1  // 配置SCL为输入上拉
15
16
17 void IIC_GPIO_Init(void);
18 void IIC_Sleep(void);
19 void IIC_Start(void);
20 void IIC_Stop(void);
21 unsigned char IIC_Wait_Ack(void);
22 void IIC_Ack(void);
23 void IIC_NAck(void);
24 void IIC_Send_Byte(unsigned char txd);
25 unsigned char IIC_Read_Byte(unsigned char ack);
26 unsigned char AT581x_ReadOneByte(unsigned char address);
27 void AT581x_WriteOneByte(unsigned char address,unsigned char data);
28
29 #endif
```

至此，软件需要适配的部分完成。

2.2、代码使用

1、TARGET_CONFIG.H 中定义了常用的 AT581x 参数，只需配置该文件的参数即可

```
#ifndef TARGET_CONFIG_H
#define TARGET_CONFIG_H

/*=====参数配置=====*/
#define TRUE 1
#define FALSE 0

#define PWR_40UA_SWITCH TRUE // 配置功耗 TRUE - 40uA; FALSE - 60uA
#define SELF_CHECK_TIME 2000 // 上电自检时间 0~65536 ms
#define LIGHT_OFF_TIME 1000 // 关灯保护时间, 建议1000ms
#define LIGHT_ON_BASE_TIME 500 // 保持默认500ms
#define LIGHT_ON_TIME 1500 // 总开灯时间=LIGHT_ON_BASE_TIME+LIGHT_ON_TIME
#define DELTA 200 // delta值: 0~1023, 值越大距离越短
#define GAIN 0x3B // 增益值范围: [0x0B, 0x1B, 0x2B, 0x3B, 0x4B, 0x5B, 0x6B, 0x7B, 0x8B, 0x9B, 0xAB, 0xBB, 0xCB]
// default = 0x3B 建议默认。近距离感应场景(2m内)使用, 如智能门锁应用, 可配置为0x7B ~ 0x9B

// 以下是光敏配置功能
#define LIGHT_SENSOR_STATUS 0 // 0-关闭光敏 1-开启光敏
#define LIGHT_SENSOR_VALUE_LOW 500 // 光敏阈值: 0~1023, 10bit
#define LIGHT_SENSOR_VALUE_HIGH 530 // 光敏阈值: 0~1023, 10bit
#define LIGHT_SENSOR_INVERSE 0 // 默认值0, 不要改动

/*=====*/
#endif
```

各个参数取值及功能均在文件中做了说明，用户也可以参考《AT58LP1T1RD 寄存器读写说明 V1.2.pdf》。

2、API 调用

```
133 /*
134 AT581x初始化
135 */
136 void AT581x_REG_Init(void)
137 {
138     set_freq(FREQ_0X5F_5869MHZ, FREQ_0X60_5869MHZ); // 如不需要重设频点, 可以不调用
139
140     set_distance(PWR_40UA_SWITCH, DELTA, GAIN);
141
142     set_light_on_time(LIGHT_ON_TIME);
143
144     set_self_check_time(SELF_CHECK_TIME);
145
146     set_light_on_base_time(LIGHT_ON_BASE_TIME);
147
148     set_light_off_time(LIGHT_OFF_TIME);
149
150     // 光敏功能, 需要时再开启
151     // set_light_sensor_threshold(LIGHT_SENSOR_STATUS, LIGHT_SENSOR_VALUE_LOW, LIGHT_SENSOR_VALUE_HIGH, LIGHT_SENSOR_INVERSE);
152
153     AT581x_WriteOneByte(0x55, 0x04);
154
155     soft_reset();
156 }
```

TARGET_CONFIG.H 中配置的参数会在 AT581x.c 中进行调用，如上图。

void AT581x_REG_Init(void)完成 AT5815 上电第一次配置。在后续程序运行过程中，如需要更新参数，可以参考 void AT581x_REG_Init(void) 或在 void AT581x_update_param(void)进行 API 调用。配置完成后后需要调用 soft_reset();使参数生效。

3、main()流程

```
21  /*-----*/
22  * 函数名: main
23  * 功能: 主函数
24  * 输入: 无
25  * 输出: 无
26  /*-----*/
27  void main()
28  {
29      DelayMs(100);
30      POWER_INITIAL(); //系统初始化, 根据用户实际MCU进行初始化
31      IIC_GPIO_Init(); // 初始化IIC
32
33      DelayMs(100);
34      AT581x_REG_Init();
35
36      while(1)
37      {
38          set_distance(0, 1023, 0x3B);
39          DelayS(20);
40          set_distance(0, 50, 0x3B);
41          DelayS(20);
42      }
43  }
```

程序初始化完成后, 会周期性设置 AT581x 的 delta 值, 可用于测试代码是否移植成功。当 delta=1023 时, 雷达在此期间不会感应, 此时运动不会触发雷达输出; 当 delta=50 时, 雷达可以感应运动并输出感应信号。

4、AT5812 与 AT5815 的使用区别

```
void set_distance(unsigned char pwr_40uA_switch, unsigned int delta, unsigned char gain);
```

设置距离的 API: set_distance 中功耗设置仅对 AT5815 芯片有效。

当使用的芯片为 AT5812 时, 功耗无法设置, 工作电流都维持在 9mA 左右。在调用 set_distance 时需要将参数 pwr_40uA_switch 默认设置为 FALSE。

3、雷达参数调试说明

本节主要对第 2 节中的一些配置参数或 API 接口进行进一步说明，并提供一些实际调试过程中的经验。

3.1、软件复位

雷达芯片的一些寄存器的值在写入后需要软件复位才能生效。当使用软件复位功能时，雷达芯片内部会重新载入寄存器的值，并使用新的参数进行工作。

在雷达芯片复位期间，雷达模块 OUT 脚会输出低电平。复位完成后，模块重新进入自检，自检期间模块 OUT 输出高电平。

在调用软件复位指令时，如模块此时处于感应状态（OUT=1），则从示波器上可以看到 OUT 脚会有一个短暂的拉低脉冲。

修改自检时间需要软件复位后才能生效。

3.2、工作功耗

AT5812 的功耗是不可设置的，3.3V 供电时工作电流约 9mA 左右。

AT5815 的功耗可以设置，3.3V 供电时，每次上电后默认的工作电流为 68uA 左右，可以通过 API 配置为最低 40uA。

用户在使用功耗设置指令时，请先和我司确认芯片的型号或通过芯片丝印区分：AT5812 的丝印为 58MD，AT5815 的丝印为 58LD。

AT5815 只能工作在电池供电的场景，不建议在 AC 转 DC 的供电场景下使用，容易受到干扰。

在其他参数完全相同的前提下，AT5815 设置为 40uA 工作电流后的极限感应距离会相对 68uA 的工作电流的距离近一些，且在最远处的感应精度也差一些。但在近距离的感应精度上没有差别。如智能门锁感应距离一般在 0.8-1.5m 左右，所以都使用 40uA 工作电流。

3.3、感应时间

感应时间指的是雷达模块检测到移动物体后 OUT 脚输出高电平的持续时间。如设置感应时间为 2s，则模块感应后会输出 2s 的高电平，但如果在输出 2s 高电平的期间雷达还持续检测到有移动物体，则 2s 的输出时间会被不断刷新，也就是顺延。

值得注意的是，感应时间的总时长=芯片内部基底输出时间+感应输出时间。其中基底输出时间默认设置为 500ms；感应输出时间可以通过指令修改，但不能设置为 0ms。也就是说，当通过指令设置感应输出时间为 1ms 时，模块感应后仍然会输出约 500ms 左右的高电平信号。

3.4、保护时间

当雷达 OUT 脚从高电平变低电平之后，即感应输出结束后，在接下来的一段时间内停止检测移动物体，这段停止检测移动物体的时间被称为保护时间。

一般来说当雷达模块的 OUT 从高变低之后，整机会做一些相应的处理或动作，如断开或闭合继电器，电机启动或停止，LED 闪烁，蜂鸣器提醒等等。这些动作一般在雷达结束输出的时候就会触发，并且这些动作会引入一些干扰并导致雷达被重新触发，最终导致模块一直被触发而持续输出高电平。所以为了避免上述情况的发生，在结束输出之后，雷达会停止检测一段时间进行保护。

保护时间默认被设置为 1s，一般不需要修改。如果某些应用在雷达结束输出后，需要控制的外设动作时间比较长而导致雷达再次被触发，则需要适当修改保护时间的长度并确保保留有足够的余量，建议留有 500ms 以上的余量。

3.5、增益

增益和感应阈值是决定雷达感应距离的最主要参数，两个参数需要合理配合以达到理想的感应效果。

雷达模块一共提供有 13 个档位增益可供设置，具体取值范围为：[0x0B, 0x1B, 0x2B, 0x3B, 0x4B, 0x5B, 0x6B, 0x7B, 0x8B, 0x9B, 0xAB, 0xBB, 0xCB]。档位越低，增益越大。理论上增益越大，可以感应的极限距离会越远，但实际应用中还需要考虑整机结构和元件影响以及模具材质带来的影响。一般来说，金属外壳的产品需要设置比较大的增益，塑料外壳增益可以低一些。

在实际调试感应距离的时候，增益参数对距离感应是一个粗调作用，比如当感应阈值固定的前提下，设置两个相邻的增益后，感应距离可能会相差 1-2 米。一般建议使用的增益范围为 0x3B~0x9B。

在径向感应的应用场景中，一般情况下增益大的参数，雷达对前方两侧的感应灵敏度会高于增益小的参数。

3.6、感应阈值

感应阈值与增益是决定雷达感应距离的最主要参数。增益实现对距离的粗调，感应阈值实现对距离的细调。

感应阈值可以调整的范围为 0~1023。由于可以调整的颗粒度很小，所以可以实现对距离的细调。感应阈值越大，感应距离越近。

一般来说，0~10 的感应阈值禁止使用，因为基本都会引起模块**自激**。11~15 的感应阈值不建议使用，因为即使模块不自激，也可能会因为**余量**不足导致后期无法大规模生产。感应阈值设置为 1023 后会导致模块无法感应。

当项目要求感应距离很长，如直线距离要求 5m 以上或挂高感应半径要求 3m 以上时，都需要对整机进行**探底**。探底可以让我们了解整机的极限性能，并可从中发现一些需要改进的地方，同时也是判断项目是否能量产的关键步骤。

关于自激、探底、余量，可以点击超链接直接跳转查看或直接翻到下方小节中查看。

3.7、光敏

觅感模块硬件上默认不焊接光敏，且某些型号上没有预留光敏电阻的硬件接口。如果

需要使用光敏功能，请先确认使用的模块上是否带有光敏硬件接口且已经焊接光敏电阻。

使用光敏功能后，只有当环境光线低到一定程度后，雷达才开启工作。

光敏参数值可以设置的范围为 0-1023，当值设置越大，则需要越低的光线才能使雷达工作。

3.8、检测窗口

检测窗口数的设置涉及雷达算法底层的检测判断，一般来说不需要也不建议修改。

检测窗口有两个参数可以调整：检测窗口数和触发窗口数。同时两个参数需要符合以下关系：检测窗口数 \geq 触发窗口数 >0 。

3.9、RF 开关

可以通过指令开启或关闭雷达芯片的 RF 电路。

模块上电默认开启 RF 电路，此时模块可以正常感应移动物体。关闭 RF 后，模块停止发送和接收射频信号，不再输出感应信号。

对于 AT5815 模块，当工作在 40uA 功耗下时，关闭 RF 后，可以进一步节省约 10uA 的功耗。

3.10、频点配置

雷达支持频点重新配置，具体配置方法可参考代码。

配置频点后，需要调用软件复位 API 使生效。

3.11、自激

自激是指在没有明显的移动物体干扰的情况下，雷达模块自己输出感应信号的现象。

自激输出的现象有多种，如不规则偶发性输出、周期性输出、持续性输出等。

造成自激的原因多种多样，下面列举一些常见导致自激的原因：

- 1、感应阈值设置过低
- 2、供电电源不稳，纹波太大
- 3、蜂鸣器、继电器等工作产生机械干扰和电源波动
- 4、二极管、三极管周期性开关，LED 闪烁等
- 5、电机、风扇、空调扇叶等工作影响

在处理自激问题时需要具体分析，有些可以通过软件处理进行规避，而有些则需要修改硬件或结构来处理。

3.12、探底

雷达模块最终是要装到整机中工作的。我们需要了解装机后的整机能够正常工作的极限参数。

获取雷达模块在整机中的极限参数的过程，称为探底。探底的目标是获取在某一个固定的增益下，让整机开始产生自激的那个感应阈值。

探底的步骤如下：

- 1、固定增益参数。
- 2、感应阈值设定一个比较大的初值，建议 200。
- 3、将整机放置在实际环境中，静置。确保环境中不要有人员或其他运动物体，观察整机中雷达是否会自激（一般观察时间为 5-10min）。
- 4、如果整机没有自激，则将感应阈值改小，重新进行步骤 3；如果整机产生自激，在说明设置的感应阈值过小，将阈值改大后，重新进行步骤 3。
- 5、不断重复上述步骤，直到找到一个感应阈值，该值刚好让整机设备处于自激的临界点，则该值就是当前增益下的探底值。

探底值可用于评估最终量产的参数值是否合理。具体评估的方法参考[余量](#)介绍。

3.13、余量

本节指的余量是在调试雷达的过程中，选择的参数应该为后面的批量生产留出足够的公差。

调试一般是在样机上进行的，所得到的参数也仅能代表雷达在调试样机上能功能，但不代表在进行批量生产的时候，所有产品的一致性都完全和样机相同。

实际的批量生产过程中，由于受到 PCB 板材、电子元器件误差、批次以及装配过程的影响，很难确保每一个 PCBA 和整机的性能都一模一样。所以如果雷达参数选择的都是比较极端的值，那么可能会导致同一批生产出来的产品有些可以正常功能而有一些功能异常，一致性不好。

一般来说，在生产中产品的性能大部分会和实验室中调试的样机基本一致，但也存在少数的产品性能会比样机差，同理也存在性能比样机好的产品。

因此，我们就需要有一个方法来告诉我们当前选定的雷达参数能否用在大批量生产中，使得该参数在性能比较差的那部分产品上也能使用。

一般来说，我们认为参数值的选择至少是探底值的 1.5~2 倍的时候，就基本可以确保该参数可以用于批量生产，也就是要留有 0.5~1 倍的余量来确保最终产品的一致性。

举个例子，假设某个整机探底值为 40，那么我们在选择最终的感应阈值的时候，最少要确保该值大于 $40 \times 1.5 = 60$ ，理想情况下应该大于 $40 \times 2 = 80$ 。如果该项目通过调试发现要使用感应阈值为 50 才能满足感应距离的要求，很明显 50 低于最低的余量要求，需要仔细分析是否还有机会通过降低整机的探底值来增加余量或者通过缩短感应距离使感应阈值增大到 60 以上来保证余量。

4、版本信息

版本号	修改内容	修改人	日期
V1.0	1、代码说明初版	Job.Huang	2021-3-1
V2.0	1、增加设置光敏 API 接口 2、增加 RF 控制 API 接口 1、增加窗口检测阈值配置 API 接口	Job.Huang	2021-9-15
V3.0	1、main 函数中增加周期性开启和关闭雷达感应的参考代码，可用于测试代码移植是否成功。	Job.Huang	2021-12-31
V4.0	1、增加雷达参数调试说明	Job.Huang	2022-2-17
V5.0	1、修改部分说明	Job.Huang	2022-4-11
V6.0	1、增加 RF 频点配置说明	Job.Huang	2022-6-22