

Telink

应用文档

泰凌定位方案介绍

AN-19052701-C7

Ver1.8.0

2022.05.31

Keyword

AOA, AOD, 恒定频率扩展

Brief

本文档为泰凌微电子定位解决方案的简要介绍。

Published by
Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,
Zhangjiang Hi-Tech Park, Shanghai, China

© Telink Semiconductor
All Rights Reserved

Legal Disclaimer

This document is provided as-is. Telink Semiconductor reserves the right to make improvements without further notice to this document or any products herein. This document may contain technical inaccuracies or typographical errors. Telink Semiconductor disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein.

Copyright © 2022 Telink Semiconductor (Shanghai) Co., Ltd.

Information

For further information on the technology, product and business term, please contact Telink Semiconductor Company www.telink-semi.com

For sales or technical support, please send email to the address of:

telinksales@telink-semi.com

telinksupport@telink-semi.com

修订历史

版本	修改内容
V1.0.0	初次发布
V1.1.0	更新第 3 章节
V1.2.0	更新 4.3.1 小节; 更新 4.3.3 小节
V1.3.0	更新第 4 章节
V1.4.0	更新第 3 章节
V1.5.0	更新 RSSI; 更新 ANT 切换顺序
V1.6.0	更新参考设计 3
V1.7.0	更新 8278 芯片
V1.8.0	更新 demo

Telink Semiconductor

Contents

修订历史	3
1 AOA/AOD	6
1.1 到达角 (AOA) 方法	6
1.2 离开角 (AOD) 方法	6
2 数据包格式	8
2.1 恒定频率扩展	8
2.2 天线切换	9
2.3 IQ 采样	9
2.4 RSSI	9
3 电缆测试	10
3.1 天线切换完整性	10
3.2 IQ 样本动态范围	10
3.3 IQ 样本一致性	10
4 硬件	12
5 固件和 API	15
5.1 API 和原始数据	15
5.2 Demo	15
5.2.1 Demo 参数	16
5.2.2 用户指南	23
5.3 泰凌解决方案发布	25
5.3.1 使用方法	26
5.3.2 连续模式	27
5.3.3 保存模式	27
5.3.4 数据包格式	28
6 现场测试结果	33
6.1 参考板 1 的现场测试结果	33
6.2 参考板 2 的现场测试结果	34
6.3 参考板 3 的现场测试结果	36
7 参考板原理图	40

List of Figures

1.1	到达角方法	6
1.2	离开角方法	7
2.1	数据包格式	8
2.2	恒定频率扩展结构	8
3.1	动态范围结果	10
3.2	不同天线的相位	11
4.1	参考板设计 1	12
4.2	参考板设计的角度说明 1	12
4.3	参考板设计 2	13
4.4	参考板设计的角度说明 2	13
4.5	参考板设计 3	14
5.1	完成收包并对收包数据进行处理	23
5.2	将 BDT 更改为 USB log 模式	24
5.3	进入日志窗口界面	24
5.4	使用 UART 进行打印的代码	25
5.5	AOA/AOD 工具界面 1 (仅与 8258 的硬件兼容)	25
5.6	AOA/AOD 工具界面 2	26
5.7	Tscript 界面	26
5.8	连续模式 (黄色: 高度角; 黑色: 水平角度)	27
5.9	AoA 接收端时隙图	27
5.10	测试数据	28
5.11	收包格式	28
5.12	三角板的数据包示例	29
5.13	多边形板的数据包示例 1	30
5.14	多边形板的数据包示例 2	31
5.15	多边形板的数据包示例 3	31
5.16	三角板 (RSSI) 的数据包示例	32
6.1	参考板 1 的测试环境	33
6.2	5 米室内测试结果	33
6.3	误差绝对值均值 vs 实际角度	34
6.4	参考板 2 的测试环境	35
6.5	估算 alpha vs 实际 alpha	36
6.6	估算 theta vs 实际 alpha	36
6.7	参考板 3 的测试环境	37
6.8	场景 1、2、3 估算 alpha vs 实际 alpha	38
6.9	场景 1、2、3 估算 theta vs 实际 alpha	38
6.10	场景 4、5、6 估算 alpha vs 实际 alpha	39
6.11	场景 4、5、6 估算 theta vs 实际 alpha	39
7.1	参考板原理图第 1 部分	40
7.2	参考板原理图第 2 部分	41

1 AOA/AOD

Telink 定位解决方案支持 Core 5.1 中定义的到达角 (Angle of Arrival, AOA) /离开角 (Angle of Departure, AOD) 功能。客户端设备在下文中为“想要获得方向信息的低功耗射频”。服务器设备在下文中为“设置方向信息基点的低功耗射频”。客户端设备可以通过 AOA/AOD 方法获得服务器设备的方向信息。客户端射频可以通过来自多个服务器设备的方向信息和提供其位置的配置文件级别信息，计算出自己的位置。

Telink 定位解决方案可以通过其嵌入式 MCU 提供估算角，同时提供 3 种天线阵列参考设计，包括 PCB 天线。

1.1 到达角 (AOA) 方法

客户端设备可以通过使用单个天线发送启用测向的数据包，使其到达角信息可用于服务器设备。

由射频开关和天线阵列组成的服务器设备，在接收这些数据包的同时切换天线并捕获 IQ 样本。IQ 样本可用于计算使用天线阵列的不同元素接收到的无线电信号中的相位差，进而可用于估算角。

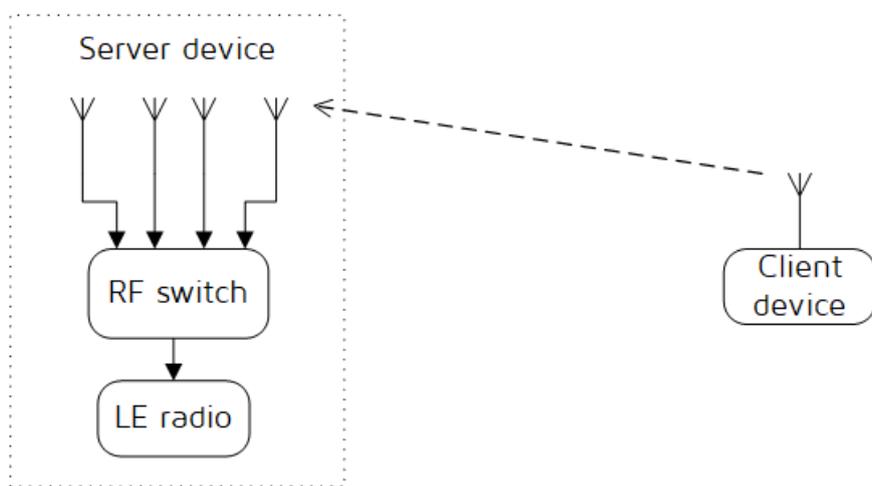


Figure 1.1: 到达角方法

1.2 离开角 (AOD) 方法

由射频开关和天线阵列组成的客户端设备，可以通过发送启用测向的数据包，在传输过程中切换天线来检测其离开角。

服务器设备接收这些数据包并捕获 IQ 样本。IQ 样本可用于计算使用天线阵列的不同元素接收到的无线电信号中的相位差，进而可用于估算角。

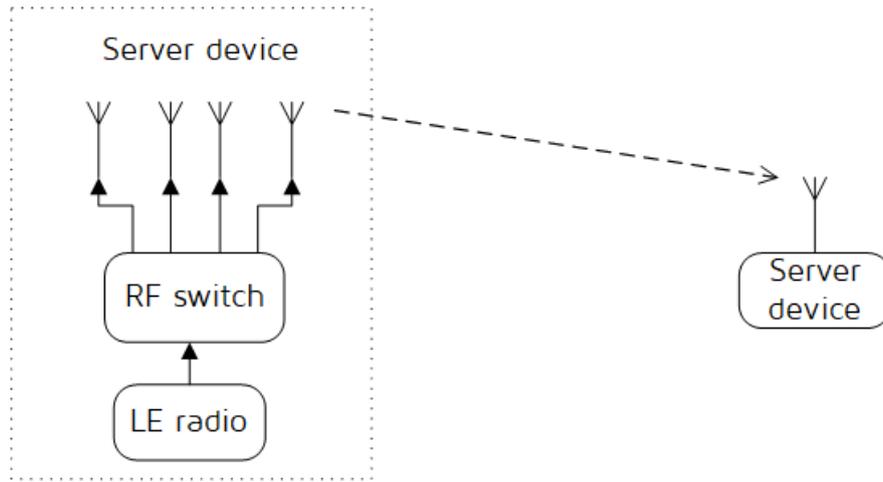


Figure 1.2: 离开角方法

Telink Semiconductor

2 数据包格式

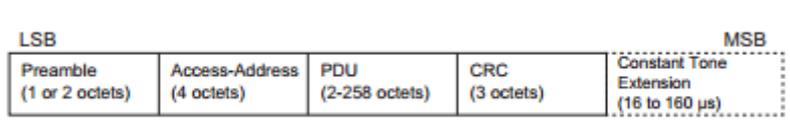


Figure 2.1: 数据包格式

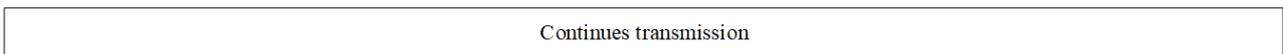
Telink 定位解决方案支持针对 AOA 和 AOD 功能指定的数据包格式：由 Preamble、Access-Address、PDU、CRC（循环冗余校验）和 Constant Tone Extension（恒定频率扩展）组成。PDU 中的 Header 包含 AOA AOD 的控制信息。

2.1 恒定频率扩展

恒定频率扩展是 AOA AOD 功能特有的。恒定频率扩展具有可变长度：至少为 16 μs 且不大于 160 μs。

恒定频率扩展的前 4 μs 被称作保护周期，接下来的 8 μs 被称作参考周期。在参考周期之后，恒定频率扩展由一系列交替的切换时隙和采样时隙组成，由主机指定每个时隙为 1 μs 或 2 μs。恒定频率扩展的结构如下所示。

AOA transmitter



AOA receiver: 1us switch slot and sample slots



AOA receiver: 2us switch slot and sample slots



AOD transmitter: 1us switch slot and sample slots



AOD receiver: 1us switch slot and sample slots



AOD transmitter: 2us switch slot and sample slots



AOD receiver: 2us switch slot and sample slots



Figure 2.2: 恒定频率扩展结构

2.2 天线切换

当设备在接收 AoA 恒定频率扩展或在发送 AoD 恒定频率扩展时，需要在天线之间进行切换，切换操作发生在称作切换时隙的时间段内。恒定频率扩展的前 4 μs 被称作保护周期，接下来的 8 μs 被称作参考周期。接收链路层在参考周期和称作采样时隙的时间段内捕获 IQ 样本。

模式中的第一根天线将在参考周期使用。模式中的第二根天线将在第一个采样时隙期间使用，第三根天线将在第二个采样时隙期间使用，依此类推。相同的天线 ID 可能在模式中出现多次，使用中的天线只会在保护周期和切换时隙期间发生变化。

2.3 IQ 采样

当接收到包含 AoD 恒定频率扩展的数据包时，接收器不需要切换天线。

当接收到包含 AoA 恒定频率扩展的数据包时，接收器将按照该速率执行天线切换，并遵循主机配置的切换模式。

接收器在参考周期内每微秒采集一个 IQ 样本，每个采样时隙进行一个 IQ 采样（因此，将有 8 个参考 IQ 样本），对于 2 μs 时隙的情况，将有最多 37 个 IQ 样本，对于 1 μs 时隙的情况，将有最多 74 个 IQ 样本。控制器将 IQ 样本报告给主机。无论其长度如何，接收器都将会对整个恒定频率扩展进行采样，除非它会与其他活动发生冲突。

2.4 RSSI

接收器用于接收数据包主体的天线上测量接收到数据包的 RSSI（这两种情况均不包括任何恒定频率扩展）。

3 电缆测试

我们根据规范“RF-PHY.TS.5.1.1”测试 8278 芯片。8278 通过了所有 AOA/AOD 测试用例。以下是有关这些测试用例的一些详细信息。

3.1 天线切换完整性

芯片 8278 实现了 3 个 GPIO 端口来控制 PA 转换，切换模式可由软件配置。8278 最多支持 8 根天线，16 个非循环切换。

3.2 IQ 样本动态范围

为了实现 MUSIC 等高阶算法，IQ 样本应显示出与接收信号强度相对应的不同幅度。规范中的测试用例：将信号 -52、-49、-57 和 -62dBm 发送到天线 1、2、3 和 4，要求 $MEAN\ ANT4 < MEAN\ ANT3 < MEAN\ ANT1 < MEAN\ ANT2$ ，“MEAN ANT n”指天线 n 的 IQ 样本幅度平均值。

芯片 8278 以 8 位精度输出 IQ 样本。下图显示了平均值和标准差 (STD)。不同端口的，逐采样点的 IQ 值，由参考信号进行归一化处理，幅度分辨率小于 1dB。

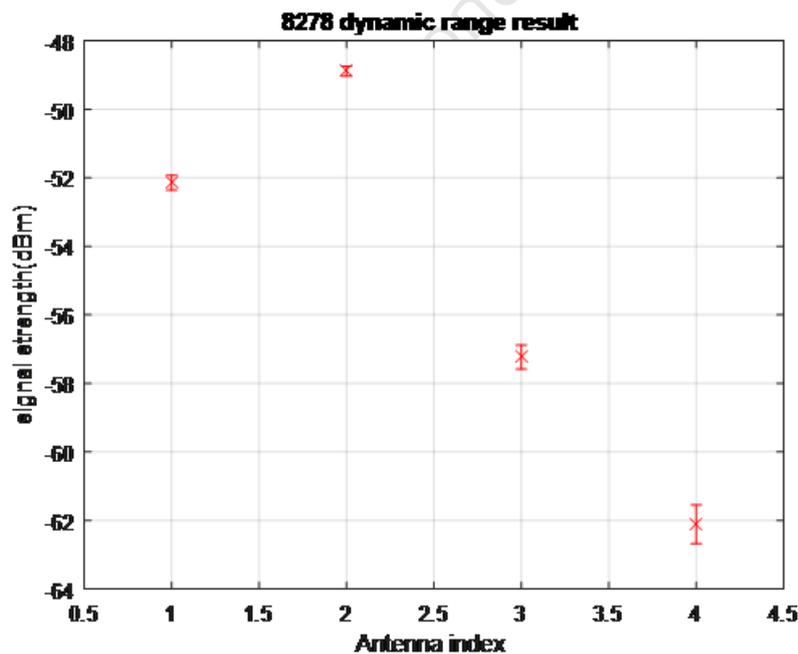


Figure 3.1: 动态范围结果

3.3 IQ 样本一致性

为了计算不同天线发射信号的相位差来获得角度，同一天线在不同时刻的相位应该是一致的，不同天线在不同时刻的相位差应该是一致的。详细信息请参阅 Core 5.1 的 5.2 章节。

使用功分器将信号传输到芯片 8278 的 4 个端口，所有数据包均附有 160 μ s CW，校准电缆差和功率分流差，用参考信号校准初始相位差。规范要求同一天线的相位变化（v-MRP）应小于 0.52，而不同天线之间的相位变化（MRPD）应小于 1.125。8278 收集的数据包中不同端口的相位标准差低于 0.025 rad。下图是 4 个端口的 IQ 样本的相位。

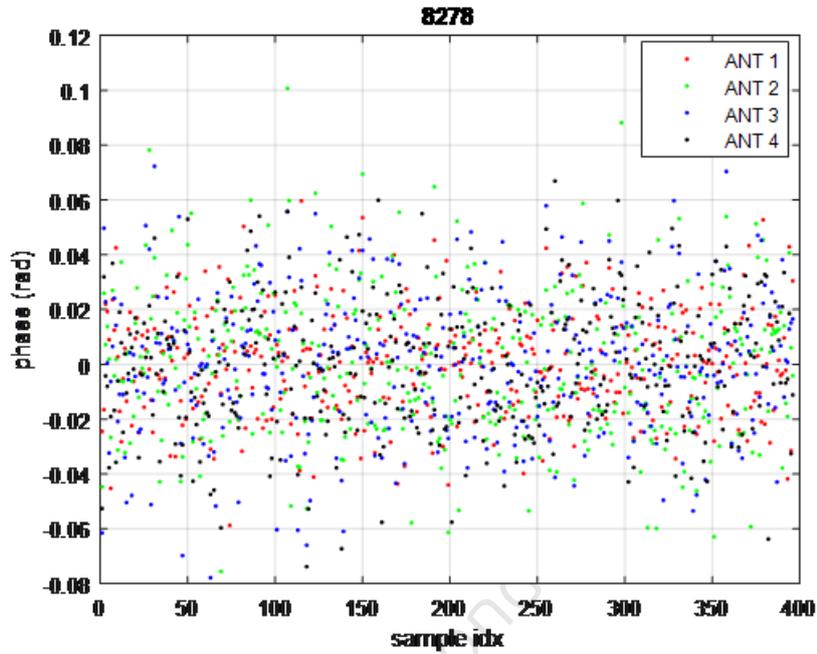


Figure 3.2: 不同天线的相位

4 硬件

参考设计有 3 种选择，参考设计 1 适用于墙上的板子，参考设计 2 适用于天花板上的板子，参考设计 3 使用 PCB 天线来替代参考设计 2 中的外部天线。

参考板设计 1 如图 4.1 所示，图 4.2 是角度说明，X 轴指向 0 度，Y 轴指向 90 度。

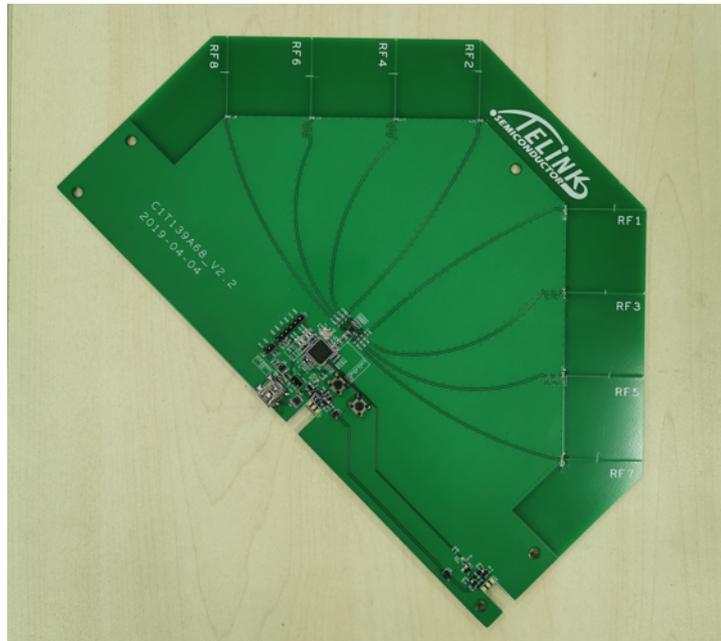


Figure 4.1: 参考板设计 1

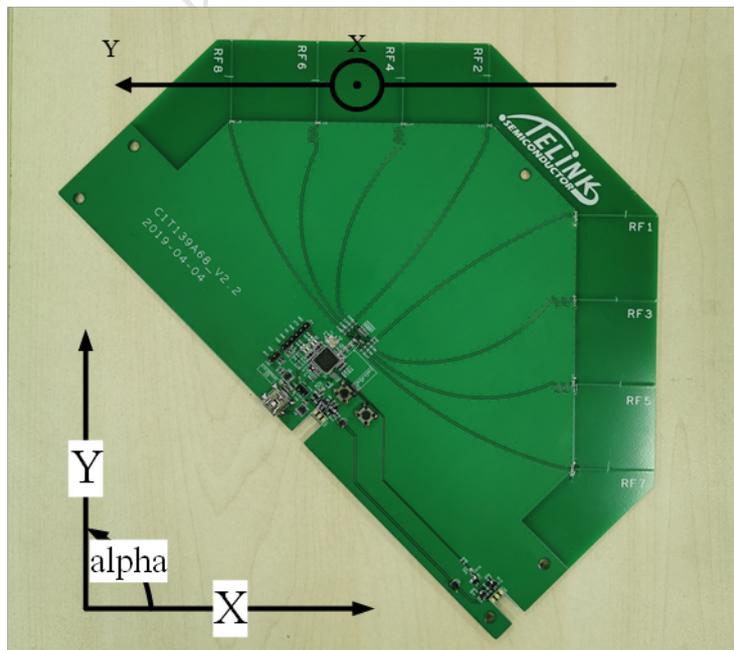


Figure 4.2: 参考板设计的角度说明 1

参考板设计 2 见图 4.3，角度说明见图 4.4，RF8-RF1 作为 X 轴指向 0 度，RF5-RF4 作为 Y 轴指向 90 度。

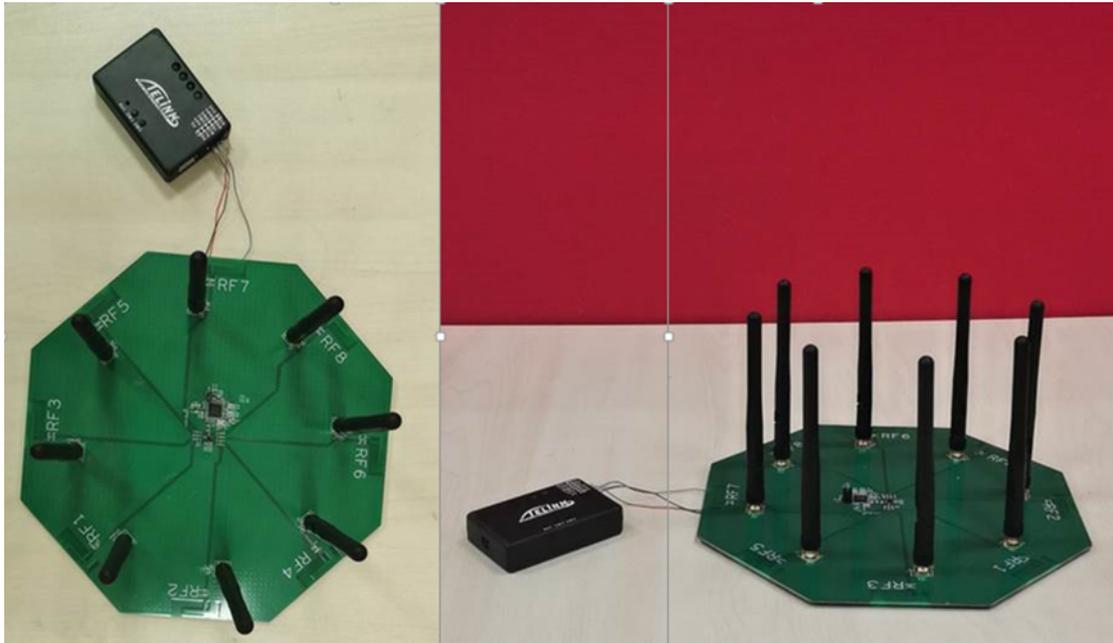


Figure 4.3: 参考板设计 2

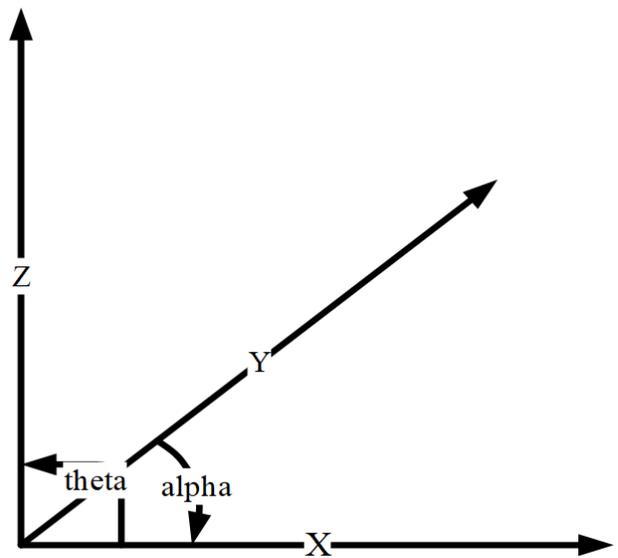
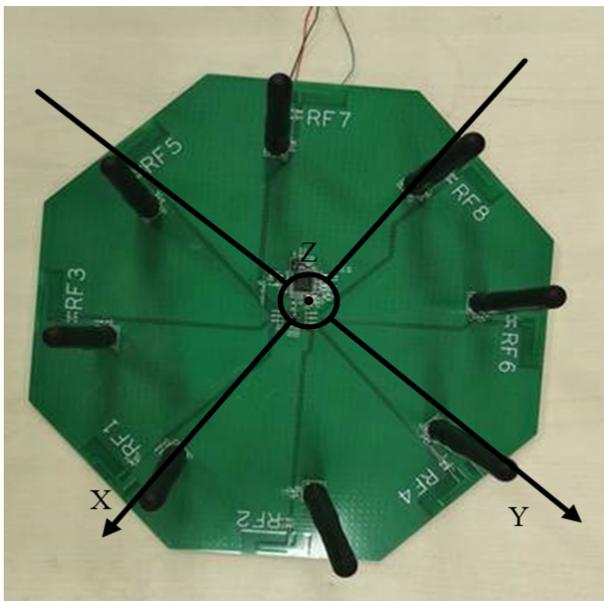


Figure 4.4: 参考板设计的角度说明 2

参考板设计 3 如图 4.5 所示，角度说明与参考板 2 相同。

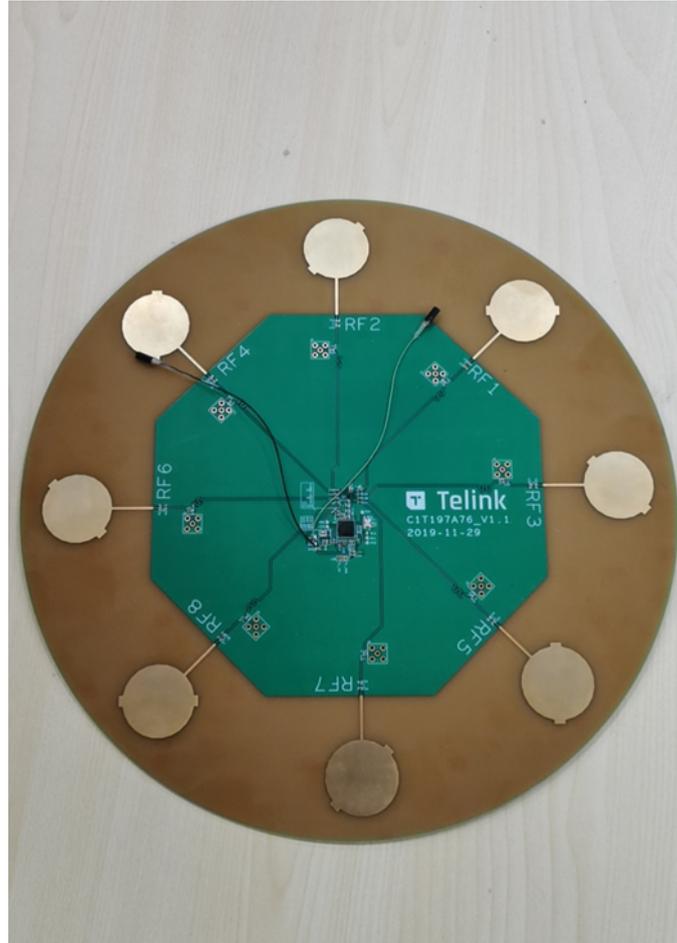


Figure 4.5: 参考板设计 3

5 固件和 API

为了方便客户使用，我们提供了用于计算角度和原始数据的 API。客户可以选择使用我们提供的 API（通过我们提供的算法）直接获取输入角度或者通过原始数据来自自己开发算法进行角度计算。同时我们提供了 demo 来帮助客户进行 AOA 或者 AOD 相关演示和功能验证。客户可以通过下面的链接获取对应的 SDK 和 Tscript 工具。

SDK: http://wiki.telink-semi.cn/tools_and_sdk/Driver/telink_b85m_driver_sdk.zip

Tscript: http://wiki.telink-semi.cn/tools_and_sdk/BLE/827x_AOA_AOD_Solution.zip

5.1 API 和原始数据

目前我们是将 AOA/AOD 收发相关配置的接口统一放在了 rf_drv.h 和 rf_drv.c，而与数据拆分及角度计算等算法相关接口放在 aoa.c 和 aoa.h 中。因此客户在使用时需要将对应的 libdriver.a 以及 rf_drv.h、aoa.h 移植到对应的工程中。用户在使用时可以在配置完 RF 初始化后通过调用 rf_drv.h 相关接口，设置使能 AOA/AOD 功能以及配置天线数量、切换顺序、采样间隔等。具体设置可参照 Demo 部分结合自身需求进行调整。另外在 aoa.c 和 aoa.h 中提供了通过收包数据计算角度信息的 API 和获取原始数据的接口。

用户可以通过调用 unsigned int raw_data_to_angle_no_z(unsigned char ptr_packet) 获取输入角度信息，该函数返回值为平面角度信息。也可以通过调用 unsigned int raw_data_to_angle_with_z(unsigned char ptr_packet, unsigned char *temp_theta) 来获取角度信息，该函数除了返回一个平面角度信息外，还会将高度角信息输出到 temp_theta 变量，用户可通过该变量获取高度角信息。示例如下：

```
unsigned int angle_value;
unsigned char theta_angle;
angle_value = raw_data_to_angle_no_z(&rx_packet[0]); //仅获取平面角度信息
angle_value = raw_data_to_angle_with_z(&data_no_amplitude[0], (unsigned char *)&theta_angle); //获取平面角度和高度角
```

另外提供了获取原始数据的接口，用户可以通过该接口获取收包端获取到的 IQ 原始数，原始数据经过数据拆分，转换为有符号十进制数，等预处理后可以提供给算法进行角度的计算。

```
unsigned char raw_data[90];
get_raw_data(&raw_data[0], &rx_packet[0], 90);
```

将原始数据复制到 raw_data[0]、raw_data[1]、raw_data[2].....raw_data[89]，它们分别代表 IQ0 的实部、IQ0 的虚部、IQ1 的实部.....IQ44 的虚部。用户需要一个至少有 90 个参数的 unsigned char 表，因为有 45 个样本数据，每个数据都包括实部和虚部。

5.2 Demo

Demo 支持 AOA/AOD 模式，客户可以通过对应的宏定义选择自己使用的模式。下载 demo 程序到参考硬件，发送端会持续向接收端发送数据以输出角度。输出的角度可以通过 UART 打印，USB 输出到 BDT/Tscript 或者 SWS 上传到 Tscript。参考硬件设计 AOA 模式的角度输出只支持 SWS 上传到 Tscript 进行输出和图形化显示。

目前 demo 兼容芯片 8278 和 8258，下面介绍 demo 时将以 8278 为例。下文中的 82xx 可根据实际情况用 8258 或者 8278 进行代替。

5.2.1 Demo 参数

drivers/82xx/aoa.h 中的变量参数:

```
#define RF_REAL_FREQ 2450 //设置 TX/RX 频率, 这里是 2450MHz。
```

定义了发送频率。

app.c 中的变量参数:

以 8278 为例:

```
/****** 选择 IO 作为天线控制脚 *****/
rf_ant_pin_sel_t ant_config =
{
    .ant_sel0_pin = RF_ANT_SEL0_PD6,
    .ant_sel1_pin = RF_ANT_SEL1_PB0,
    .ant_sel2_pin = RF_ANT_SEL2_PB1,
};
```

目前 Demo 兼容 8258 和 8278, 因此分别提供了一组 IO 作为天线控制脚。

```
/****** 选择 UART 输出或者 Tscript 输出 *****/
#define BY_TSCRIPT      1
#define BY_UART        2
#define USER_MODE      BY_TSCRIPT
```

采用 UART 或者 Tscript 输出角度信息。

```
/****** 选择 TX 或 RX *****/
#define TX      1
#define RX      2

#define RF_TXRX_MODE      RX
```

将电路板定义为发送端或接收端。

```
/****** 选择 AOA 或 AOD *****/
#define RF_AOD_EN      0 //0: AOA; 1: AOD
```

将电路板定义为 AOA 或 AOD。

```
#define calibration_no_Z      0 //0: 采用返回带高度角信息的算法; 1: 使用只返回平面角信息的算法
```

决定采用的算法处理结果是否带有高度角信息。

8258 芯片可调用如下函数:

```

/***** 选择天线数量和切换顺序 *****/
/**
 * @brief 以 4 根天线为例说明天线切换顺序
 *      SWITCH_SEQ_MODE0    - 天线索引切换顺序 01230123
 *      SWITCH_SEQ_MODE1    - 天线索引切换顺序 0123210
 */
rf_aoa_aod_ant_init(8,&ant_config,SWITCH_SEQ_MODE0);
    
```

注意:

- 8258 仅支持两种天线切换顺序，这部分可以参照 rf_drv.h 中两种天线切换模式的注释。

8278 芯片则可参照如下设置:

```

unsigned char antenna_switch_seq[8] = {0,1,2,3,4,5,6,7};
rf_aoa_aod_ant_init(8,&ant_config,SWITCH_SEQ_MODE0,antenna_switch_seq);
    
```

8278 在天线切换设置上相较于 8258 做的更加灵活，将原来直接设置天线切换顺序的方式修改为查表的方式，通过 SWITCH_SEQ_MODE x ($x=0,1,2$) 来设置查表的顺序；通过 antenna_switch_seq 设置表内容，即实际的天线切换顺序。

```

/*****TX/RX 频率 *****/
[简要描述]: 射频频率，为了改变频率，请在 drivers/82xx/aoa.h 中更改 RF_REAL_FREQ。
/*****/
#define RF_FREQ          RF_REAL_FREQ-2400    // TX/RX 频率
    
```

```

/*****TX Payload 长度 *****/
#define TX_PKT_PAYLOAD          0x12
    
```

定义了发送数据包的 payload 部分的长度，根据包结构可以使用 payload length 进行长度校验以及获取数据包中存储信息的偏移量。

射频频率应在 aoa.h 中通过更改 RF_REAL_FREQ 进行验证，不在此更改。

```

/*****TX 能量 *****/
#if (MCU_CORE_B85)
#define RF_POWER          RF_POWER_P10p46dBm
#elif (MCU_CORE_B87)
#define RF_POWER          RF_POWER_P11p26dBm
#endif
    
```

定义了发射能量 (TX power)，这部分根据芯片的能量表设置即可。

```
#define IQ_DATA_8_BIT_EN    0 // 0: 采用 20 bit 长度的 IQ 数据; 1: 采用 8 bit 长度的 IQ 数据。
```

定义了 IQ 数据的长度，8258 仅支持 8 bit 长度的 IQ 数据；8278 目前支持 8bit、16bit(low)、16bit(high)、20bit 长度的 IQ 数据。目前 8278 部分 demo 仅以 8 bit 和 20 bit 长度数据作为示例。

```
/****** 连接接入码 *****/
#define ACCESS_CODE        0xfcaab2c1
```

定义 ACCESS_CODE.

```
/****** 脚本指令空间 *****/
unsigned char para_buff[commandBuffSize] = {0};
unsigned char temp_para[commandBuffSize] = {0};
```

当采用 Tscript 进行角度显示时，该空间用于指令传输。

```
/****** 记录 tx 或 rx cnt *****/
volatile unsigned int rx_cnt=0;//用于表示收包个数。
volatile unsigned char tx_cnt=0;//发包个数，同时会将该数据放入包中，方便收包端确认数据来自不同的包。
```

```
/****** 存储 stimer tick 值 *****/
unsigned int tick_now;//用于存储当前 tick 值以判断超时
```

```
/****** 角度信息 *****/
unsigned int angle_value;    //用于存储平面角度信息
unsigned char theta_angle;  //将高度角结果存储为带有高度角信息的输出参数
```

这两个变量会作为结果传输到 Tscript 脚本供上层显示角度使用。

```
/****** 原始数据获取空间 *****/
unsigned char raw_data[90];
```

用于存储算法使用的 IQ 原始数据，demo 中在使用 20 bit 长度的 IQ 数据时会先将其转化为 8 bit 长度的数据，然后存储在原始数据中。

```
/****** 上传结果到 Tscript *****/
unsigned char trans_buff[136] = {0x00};
unsigned char trans_table[10];
```

用于将数据上传到 Tscript。

```

/***** 收包数据 *****/
unsigned char rx_packet[280] __attribute__((aligned(4))) = {0x00}; //280 byte 空间以满足 20
↳ bit 长度的 IQ 数据, 可根据不同的长度的 IQ 数据进行调整。
unsigned char data_has_amplitude[256] __attribute__((aligned(4))) = {0};
unsigned char data_no_amplitude[256] __attribute__((aligned(4))) = {0};
    
```

用于存储从 rx_packet 处理后得到的带有高度角信息和不带高度角信息的数据, 供计算角度的算法使用。

```

/***** 天线角度校准 *****/
int phase_rx_angle_cali_angle[8] = {0, -41, 82, 163, 41, -122, 122, 122};
    
```

算法是针对 rx_angle 的校准。

User_init 介绍:

```

void user_init()
{
    gpio_init(1);
    #if (MCU_CORE_B87 || MCU_CORE_B85)
    //用于使能 usb, 可以通过 usb 输出角度信息。
    usb_set_pin_en();
    #endif
    //对 BLE 模式进行初始化, 不区分 tx, rx。
    rf_drv_init(RF_MODE_BLE_1M_NO_PN);
    #if(MCU_CORE_B85)
    /***** 天线配置 *****/
    [简要描述]:
    主要配置天线的个数, 控制天线切换的引脚, 以及天线切换的顺序。
    8: 代表使能 8 根天线, 目前 kite 最多支持 8 根天线。客户可根据需求使能相应数量。
    ant_config: 用于控制天线的引脚
    SWITCH_SEQ_MODE0: 天线切换模式
    *****/

    rf_aoa_aod_ant_init(8, &ant_config, SWITCH_SEQ_MODE0);
    /***** IQ 采样配置 *****/
    [简要描述]:
    该函数主要用于 IQ 数据长度、采样间隔和采样偏移。
    IQ_8_BIT_MODE: 8 bit 长度的 IQ 数据
    SAMPLE_AOA_4US_AOD_CTEINFO_INTERVAL: 如果是 AOA, 那么 4 us 的采样间隔就对应于协议的 2 us 作为一个
    ↳ 时隙; 如果是 AOD 模式则需要根据 CTEinfo 来判断是 4 us 采样间隔还是 2 us 的采样间隔。这部分具
    ↳ 体可以参照协议。
    -4: 该参数用来调整初始采样点的位置, 为确保采样点为最优。
    *****/

    rf_aoa_aod_sample_init(IQ_8_BIT_MODE, SAMPLE_AOA_4US_AOD_CTEINFO_INTERVAL, -4);
    #else
    
```

```

/***** 天线配置 *****/
[简要描述]:
主要配置天线的个数，控制天线切换的引脚，以及天线切换的顺序。需要注意的是 8278 相较于 8258 的天线
↪ 切换顺序的设置有所不同。
8: 代表使能 8 根天线，与 8258 含义相同
ant_config: 用于控制天线的引脚，与 8258 含义一致
SWITCH_SEQ_MODE0: 天线切换中的查表顺序，在 8258 中则为实际天线切换顺序
antenna_switch_seq: 天线切换顺序表，在表中找到对应位置所需要切换的天线
*****/

rf_aoa_aod_ant_init(8,&ant_config,SWITCH_SEQ_MODE0,antenna_switch_seq);
#if(IQ_DATA_8_BIT_EN)
/***** IQ 采样配置 *****/
[简要描述]: 同上
*****/

rf_aoa_aod_sample_init(IQ_8_BIT_MODE,SAMPLE_AOA_4US_AOD_CTEINFO_INTERVAL,-4);
#else
rf_aoa_aod_sample_init(IQ_20_BIT_MODE,SAMPLE_AOA_4US_AOD_CTEINFO_INTERVAL,-4);
#endif
#endif

/***** 算法初始化 *****/
#if calibration_no_Z
    init_lookup_table_algorithm_no_Z();
#else
    init_lookup_table_algorithm_with_Z();
#endif

/***** 发送端使能 AOA/AOD 功能 *****/
#if(RF_TXRX_MODE==TX)
    rf_aoa_aod_set_tx_mode(RF_TX_ACL_AOA_AOD_EN);
    rf_set_power_level_index (RF_POWER);
#elif(RF_TXRX_MODE==RX)
    rf_aoa_aod_set_rx_mode(RF_RX_ACL_AOA_AOD_EN);
#endif

/***** 确保芯片处于 idle 状态 *****/
    rf_set_tx_rx_off();
    rf_set_tx_rx_off_auto_mode();

/***** 设置发送接收频点及接入码 *****/
    rf_set_channel(RF_FREQ,0);
    rf_access_code_comm(ACCESS_CODE);
    
```

```

/*****Tscript 模式初始化指令和结果空间 *****/
#if(USER_MODE == BY_TSCRIPT)
    //2.initiate parameter buffer to receive CMD and parameter
    ParaBuf_Init(unsigned int)commandBuff,commandBuffSize,commandBuffCnt);
    //3.initiate result buffer to send the result
    ResuBuf_Init(unsigned int)resultBuff,resultBuffSize,resultBuffCnt);
#endif
}
    
```

Main_loop 介绍:

Tx 部分主要是使能 tx 模式，然后根据需求设置相应的发包间隔，进行 AOA/AOD 数据包的发送即可，具体可以参考如下代码。

```

/*****Tx 部分 *****/
rf_set_txmode();          //开启 tx 模式
    while(1)
    {
        while((unsigned int)(clock_time()-tick_now) < 400000); //设置超时
        tick_now = clock_time();
        tx_cnt++;
    //发送对应的数据包
    #if RF_AOD_EN
        ble_tx_packet_AOD[7] =tx_cnt;
        rf_tx_pkt (ble_tx_packet_AOD);
    #else
        ble_tx_packet_AOA[7] =tx_cnt;
        rf_tx_pkt (ble_tx_packet_AOA);
    #endif
        while(!rf_tx_finish()); //等待发包结束
        rf_tx_finish_clear_flag(); //清除中断标志
    }
    
```

RX 部分主要工作包含：收包、计算角度信息、输出角度信息。角度的输出包含通过 UART 打印和通过 Tscript 显示两种方式。其中在使用 Tscript 显示时，可以根据收包端与上位机连接是通过 single wire 还是 usb table 选择运行不同的脚本。

由于 RX 部分比较复杂，下面会根据功能分块简单介绍一下。

```

/*****Rx 部分 *****/
#elif(RF_TXRX_MODE==RX)
    rf_rx_buffer_set(unsigned char*)rx_packet,296, 0);
    rf_set_rxmode ();
    sleep_us(85);
    rx_tick = clock_time();
    //这部分的主要功能是设置收包 buffer，使能收包功能
    
```

```

while(1)
{
    #if(USER_MODE == BY_TSCRIPT)
        if(is_ParaBuf_HaveData() != 0)
        {
            ParaBuf_Read(para_buff,commandBuffSize);

            if(para_buff[0] == 0x33)
            {
                trans_table[1] = RF_FREQ;
                trans_table[2] = ACCESS_CODE&0xff;
                trans_table[3] = (ACCESS_CODE>>8)&0xff;
                trans_table[4] = (ACCESS_CODE>>16)&0xff;
                trans_table[5] = (ACCESS_CODE>>24)&0xff;
                trans_table[6] = 2;

                #if(MCU_CORE_B85)
                    trans_table[7] = 1;
                #else
                    trans_table[7] = IQ_DATA_8_BIT_EN;
                #endif

                trans_table[8] = (TX_PKT_PAYLOAD&0xff);
                ResuBuf_Write(trans_table, 32);
            }
        }
    }
}
    
```

在使用 Tscript 模式下等待脚本下达对应指令 para_buff[0]==0x33 时，上传程序的基本信息包括收发包频点，接入码等，供脚本显示信息或者处理数据使用。

```

        if(para_buff[0] == 0x55)
    #endif
    {
        while(1)
        {
            #if(USER_MODE == BY_TSCRIPT)
                if(is_ParaBuf_HaveData() != 0)
                {
                    ParaBuf_Read(para_buff,commandBuffSize);
                    if(para_buff[0] == 0x11)
                    {
                        while(is_rf_receiving_packet());
                        rf_set_tx_rx_off();
                        Result_Buff_Clear();
                        break;
                    }
                }
            }
        }
    }
    
```

```

    }
}

```

获取指令 0x55 进入收包状态，判断是否需要清理 result buffer，如需要清理等待收包结束。关闭收包状态，并清空 result buffer。

```

if(rf_is_rx_finish())
{
    if(rf_is_rx_right() && rf_aoa_aod_is_rx_pkt_len_ok(rx_packet))
    {
        rx_cnt++;
        rx_tick = clock_time();
        for(unsigned char i=0; i<rf_aoa_aod iq_data_offset(rx_packet); i++)
        {
            data_has_amplitude[i] = rx_packet[i];
            data_no_amplitude[i] = rx_packet[i];
        }
    }
}

#if(IQ_DATA_0_BIT_EN || MCU_CORE_B05)
#else
frond_end(&rx_packet[rf_aoa_aod iq_data_offset(rx_packet)], &data_has_amplitude[rf_aoa_aod iq_data_offset(rx_packet)], &data_no_amplitude[rf_aoa_aod iq_data_offset(rx_packet)]);
for(unsigned int i=0; i<9; i++)
{
    data_has_amplitude[rf_aoa_aod iq_group_number(rx_packet)*2+rx_packet[5]+6+i] = rx_packet[rf_aoa_aod_hdinfo_offset(rx_packet)+i];
    data_no_amplitude[rf_aoa_aod iq_group_number(rx_packet)*2+rx_packet[5]+6+i] = rx_packet[rf_aoa_aod_hdinfo_offset(rx_packet)+i];
}
#endif

#if calibration_no_Z
    angle_value = raw_data_to_angle_no_z(&rx_packet[0]);
#else
#endif
#if(IQ_DATA_0_BIT_EN || MCU_CORE_B05)

```

Figure 5.1: 完成收包并对收包数据进行处理

截图信息一直到 main_loop 结束这部分主要是完成收包，并对收包数据进行处理，通过算法得出角度信息。

同时会将包里面的数据得出的原始数据保存下来上传到 Tscript。客户可以通过 Tscript 的脚本获取原始数据来简单验证自己的算法。

5.2.2 用户指南

软件：

- Telink IDE 1.3
- Telink Burning and Debugging Tool (BDT)

步骤：

- 将 app.c 中的 RF_TXRX_MODE 改为 TX, 然后使用 Telink IDE 1.3 构建项目生成.\RF_AOA_Demo\RF_AOA_Demo.bin。
- 用 usb 或 SW 连接单天线板，下载 TX 程序。
- 将 app.c 中的 RF_TXRX_MODE 改为 RX, 然后重建项目, 生成一个新的.\RF_AOA_Demo\RF_AOA_Demo.bin。
- 通过 usb 或 SW 连接多天线板，下载 RX 程序。
- 使用 usb 连接多天线 RX 板供电和通信，然后通过 View -> usb log 将 BDT 更改为 usb log 模式，如下图所示。

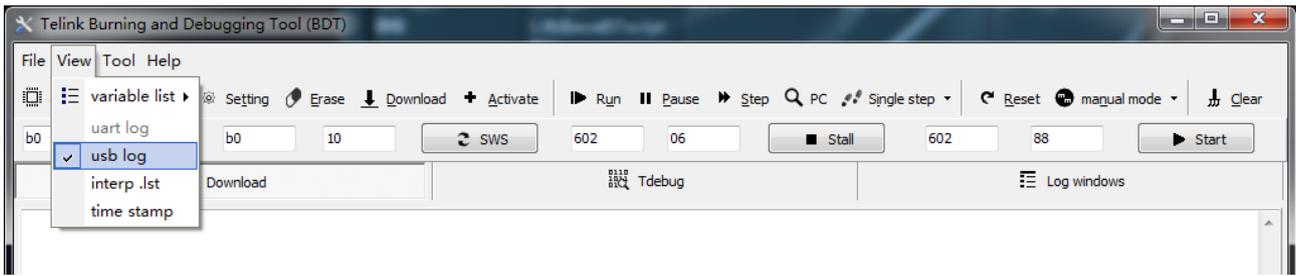


Figure 5.2: 将 BDT 更改为 USB log 模式

- f. 打开日志窗口。这样用户可以在收到 AOA 数据包后看到输入角度。如果使用程序中的打印功能，请确保将其打印出来，否则程序会出现未知的错误。

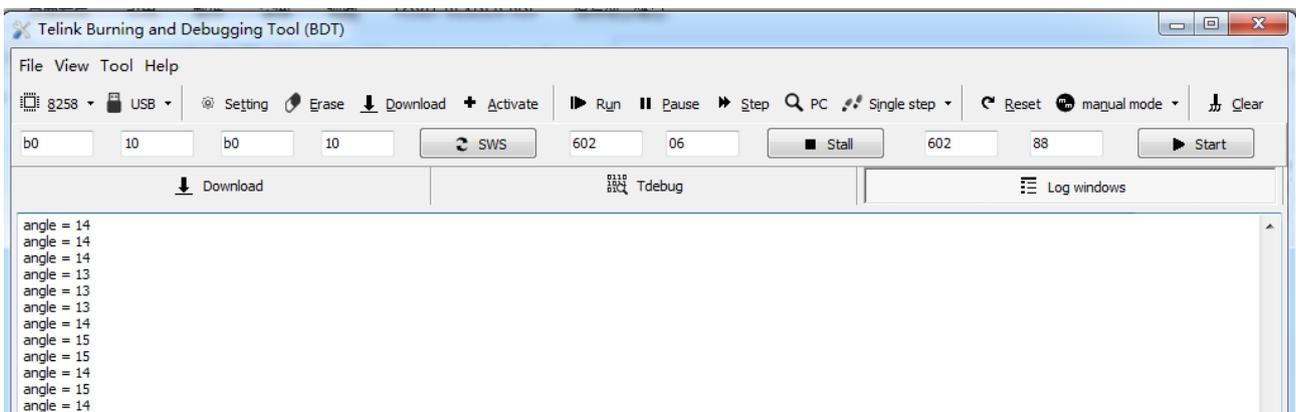


Figure 5.3: 进入日志窗口界面

- g. 如果用户想使用 UART 进行打印，需要在 drivers/8258/printf.h 中修改一些定义。将 DEBUG_BUS 改为 DEBUG_IO，设置 PRINT_BAUD_RATE 和 DEBUG_INFO_TX_PIN，然后 GPIO 将模拟 UART 并以高达 1000000 的波特率打印出信息。

```

#define DEBUG_IO          1
#define DEBUG_USB        2

#define DEBUG_BUS        DEBUG_IO

#if (DEBUG_BUS==DEBUG_USB)
/**
 * @brief      This function serves to printf string by USB.
 * @param[in] *format - format string need to print
 * @param[in] ...    - variable number of data
 * @return     none.
 */
void usb_printf(const char *format, ...);
#define printf      usb_printf
#elif (DEBUG_BUS==DEBUG_IO)
#define PRINT_BAUD_RATE      1000000 //1M baud rate
#define DEBUG_INFO_TX_PIN    GPIO_PB4

```

Figure 5.4: 使用 UART 进行打印的代码

5.3 泰凌解决方案发布

除了演示外，我们还提供了 Telink AOA/AOD 解决方案，用于显示 AOA/AOD 结果，并用所述硬件收集原始数据。解决方案包括 bin 文件和调试工具——Tscript，界面如下。

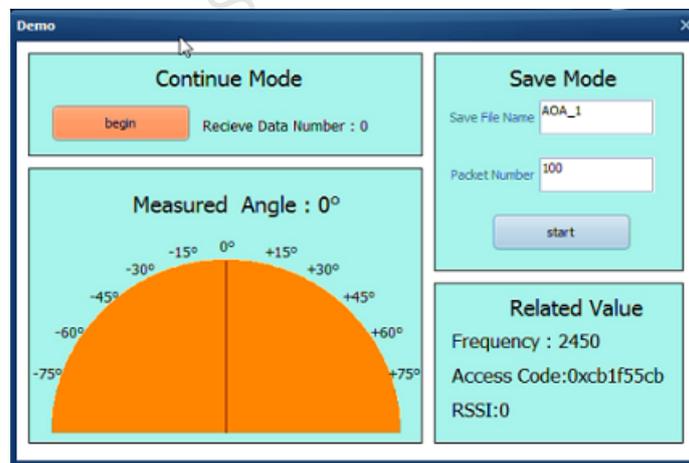


Figure 5.5: AOA/AOD 工具界面 1 (仅与 8258 的硬件兼容)

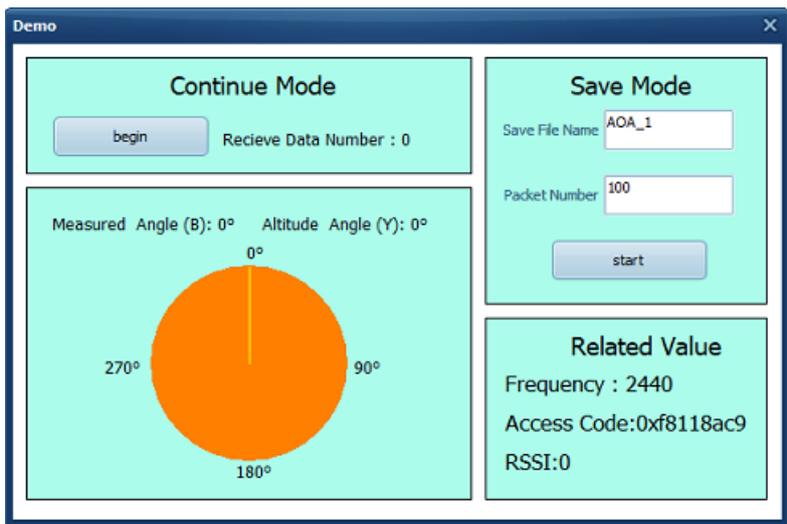


Figure 5.6: AOA/AOD 工具界面 2

5.3.1 使用方法

步骤 1: 将指定的 bin 文件下载到芯片, 然后使用 USB 或 SWS 将电路板连接到计算机以打开电源并进行通信。通过 SWS 将多边形板连接到计算机, 特别是在使用多边形板测试 AOA 时, 因为多边形板上没有 USB。

步骤 2: 打开 Tscript.exe, 双击 RF_AutoTest_Kite\AoA.lua, 然后界面如下图所示。根据电脑和主板之间不同的连接方式, 选择 AoA_Sniffer_draw_by_sws_V1.4.lua 或 AoA_Sniffer_draw_by_usb_V1.4.lua。

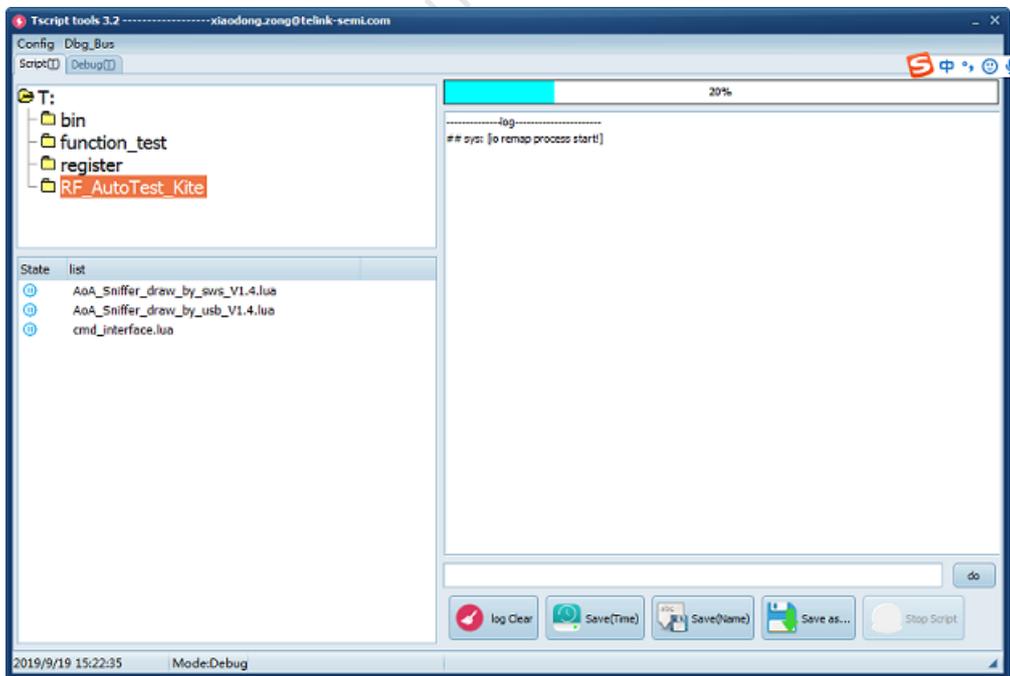


Figure 5.7: Tscript 界面

步骤 3: 选择连续模式或保存模式启动, 这两种模式将在5.3.2 连续模式和5.3.3 保存模式中介绍

5.3.2 连续模式

连续模式旨在显示内部 API 的计算角度结果。接收板收到 AOA/AOD 数据后会不断接收数据并计算角度。固定窗口的平均角度数据后，结果会显示在界面中，表中的指针会指向相应的位置。此外，接收到的数据包编号、频率、接入码和 RSSI 在界面中显示如下。当前界面支持显示高度角和水平角度。

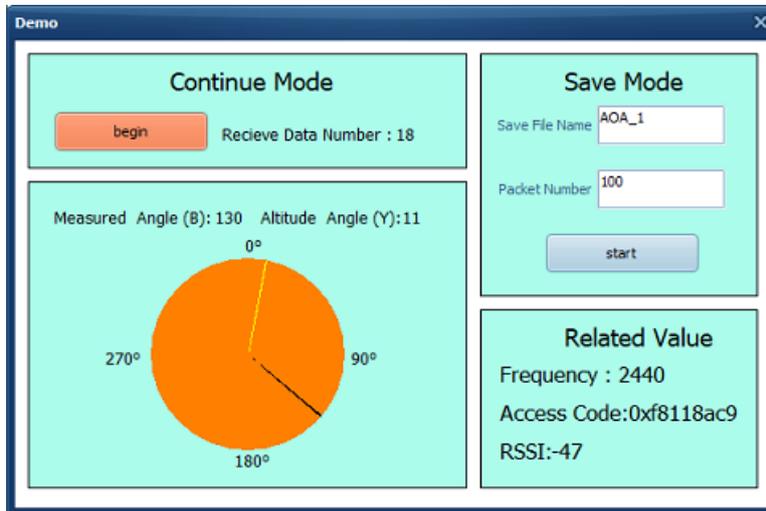


Figure 5.8: 连续模式（黄色：高度角；黑色：水平角度）

5.3.3 保存模式

保存模式旨在收集 AOA/AOD 原始数据。输入“保存文件名”和“数据包编号”后点击“开始”按钮，然后用户可以获得包含十进制采样 IQ 值的 Save_File_Name.txt 和包含十六进制原始接收数据的 Save_File_Name_backup.txt。用户可以使用其他算法来处理这些数据或进行其他实验。

Save_File_Name.txt 将包含 n 个数据包的十进制采样 IQ 值，其中 n 是用户设置的数据包数量。每个数据包包含 45 个组，包括对应于 45 个采样点信息的 90 个数字。每组包含两个数字，第一个数字表示采样点的实部，第二个数字表示采样点的虚部。下图显示了 AoA 接收端的时隙图。在参考周期内，芯片每 1 μ s 采样一次信息，共得到 8 组编号。之后，芯片将在每个采样时隙采样信息，共 37 组编号。所以每个数据包包含 37 + 8 = 45 组，共 90 个数字。

AoA receive: 2 μ s switching and sampling slots

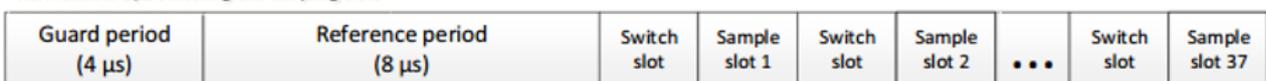


Figure 5.9: AoA 接收端时隙图

例如，上图显示一个 AOA 测试数据包，其中包含前面提到的总共 90 个数字。第一行列出了在参考周期内采样的 16 个数字，采样结果如下：

-97 - 18i, 2 - 86i, 88 - 14i, 2 + 88i, -90 + 23i, -34 - 97i, 79 - 26i, 41 + 90i

用户还可以通过这种方式获取其他 37 组采样时隙数据。

```

-97 -18 2 -86 88 -14 2 88 -90 23 -34 -97 79 -26 41 90
64 63 74 78 95 47 94 27 92 -1 94 1 72 -38 83 -62
106 -119 47 -69 -6 -91 9 -102 -24 -74 -37 -68 -100 -127 -66 -58
-89 -38 -103 -12 -98 22 -74 17 -75 39 -46 75 -49 87 -17 68
19 100 20 82 42 81 116 87 70 38 75 27 95 -6 89 -1
88 -31 79 -64 93 -114 32 -89 32 -83
RSSI :-80
Input angle :1
    
```

Figure 5.10: 测试数据

文件位于.\kite_RF_AoA\project\RF_AutoTest_Kite\AutoTest_Report。

默认传输频率为 2460 MHz，天线切换顺序为：三角板为 RF2-> RF4-> RF6-> RF8，RF2 为参考周期的主天线；多边形板为 RF1-> RF2-> RF3-> RF4-> RF5-> RF6-> RF7-> RF8，RF1 为参考周期的主天线。

5.3.4 数据包格式

这部分将介绍数据包格式并举例说明。数据包格式如下：

Packet Length (4 bytes) *1	Header 0 (1 byte)	Payload Length (1 byte)	AOA/AOD Specied data (1 byte) *2	Payload (n bytes)	Raw Data (90 bytes or 164 bytes) *3	Packet State Message (8 bytes)
-------------------------------	----------------------	----------------------------	--	----------------------	--	-----------------------------------

*1 This 4 bytes are not included in length

*2 AOA : 0x14

AOD : 0x94

*3 Switch slot 1us : 164 bytes

Switch slot 2us : 90 bytes

Figure 5.11: 收包格式

例如，这里有一个由三角板（由 Tscript 保存）接收的 2 us 切换时隙的 AOA 数据包。如果多天线板是三角板，则可以根据数据包格式进行如下分析。在前面提到的解决方案的保存模式中，一旦接收到 AOA/AOD 数据包，接收端将会调用函数来计算角度和 r 的平方，然后将其放在 rx_packet 后面。Tscript 将读取总共 130 个字节来获取它们。

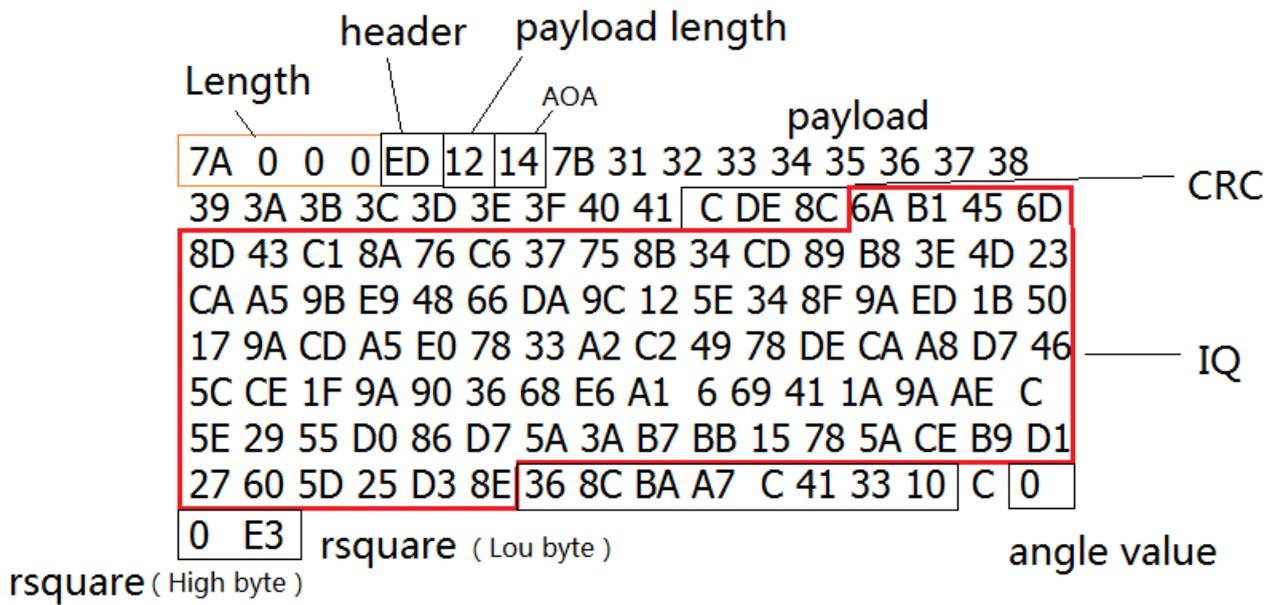


Figure 5.12: 三角板的数据包示例

rx_packet 中的原始数据需要被看作 char 变量，所以最高位是符号位，较低的 7 位是数据位，它们可以用标准方法转换为十进制数。例如，示例中的原始数据可以转换为如下图所示。天线切换顺序为 RF2-> RF4 -> RF6 -> RF8。

RF2(reference)							
Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part
6A	B1	45	6D	8D	43	C1	8A
76	C6	37	75	8B	34	CD	89
RF4		RF6		RF8		RF2	
Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part
B8	3E	4D	23	CA	A5	9B	E9
48	66	DA	9C	12	5E	34	8F
9A	ED	1B	50	17	9A	CD	A5
E0	78	33	A2	C2	49	78	DE
CA	A8	D7	46	5C	CE	1F	9A
90	36	68	E6	A1	6	69	41
1A	9A	AE	C	5E	29	55	D0
86	D7	5A	3A	B7	BB	15	78
5A	CE	B9	D1	27	60	5D	25
D3	8E						
Convert to							
RF2(reference)							
Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part
106	-79	69	109	-115	67	-63	-118
118	-58	55	117	-117	52	-51	-119
RF4		RF6		RF8		RF2	
Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part
-72	62	77	35	-54	-91	-101	-23
72	102	-38	-100	18	94	52	-113
-102	-19	27	80	23	-102	-51	-91
-32	120	51	-94	-62	73	120	-34
-54	-88	-41	70	92	-50	31	-102
90	36	68	E6	A1	6	69	41
26	-102	-82	12	94	41	85	-48
-122	-41	90	58	-73	-69	21	120
90	-50	-71	-47	39	96	93	37
-45	-114						

Figure 5.13: 多边形板的数据包示例 1

例如，这里是多边形板（由 Tscript 保存）接收的 2 μs 切换时隙的 AOA 数据包，可以根据数据包格式进行如下分析。在前面提到的解决方案的保存模式中，接收端会在收到 AOA/AOD 数据包后调用函数来计算角度，然后将其放在 rx_packet 后面。Tscript 将读取总共 130 个字节来获取它们。

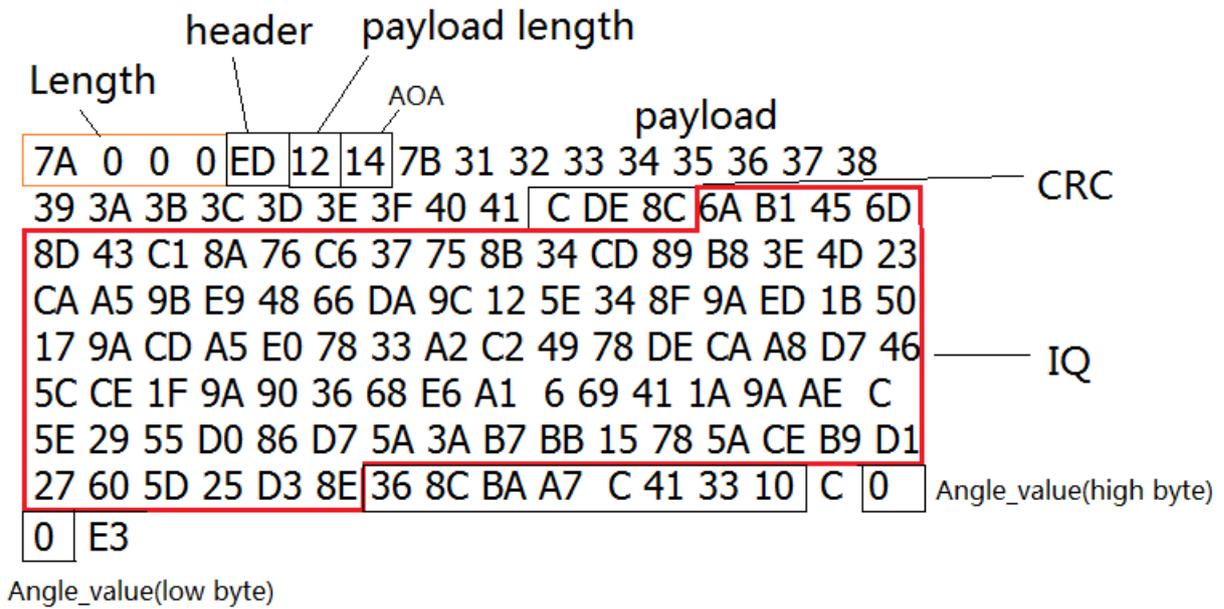


Figure 5.14: 多边形板的数据包示例 2

示例中的原始数据可以转换为如下图所示。天线切换顺序为 RF1-> RF2-> RF3-> RF4-> RF5-> RF6-> RF7-> RF8。

RF1(reference)															
Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part
6A	B1	45	6D	8D	43	C1	8A	76	C6	37	75	8B	34	CD	89
RF2		RF3		RF4		RF5		RF6		RF7		RF8		RF1	
B8	3E	4D	23	CA	A5	9B	E9	48	66	DA	9C	12	5E	34	8F
9A	ED	1B	50	17	9A	CD	A5	E0	78	33	A2	C2	49	78	DE
CA	A8	D7	46	5C	CE	1F	9A	90	36	68	E6	A1	6	69	41
1A	9A	AE	C	5E	29	55	D0	86	D7	5A	3A	B7	BB	15	78
5A	CE	B9	D1	27	60	5D	25	D3	8E						
convert to															
RF1(reference)															
Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part	Real part	Imag part
106	-79	69	109	-115	67	-63	-118	118	-58	55	117	-117	52	-51	-119
RF2		RF3		RF4		RF5		RF6		RF7		RF8		RF1	
-72	62	77	35	-54	-91	-101	-23	72	102	-38	-100	18	94	52	-113
-102	-19	27	80	23	-102	-51	-91	-32	120	51	-94	-62	73	120	-34
-54	-88	-41	70	92	-50	31	-102	90	36	68	E6	A1	6	69	41
26	-102	-82	12	94	41	85	-48	-122	-41	90	58	-73	-69	21	120
90	-50	-71	-47	39	96	93	37	-45	-114						

Figure 5.15: 多边形板的数据包示例 3

RSSI 值代表接收数据的信号强度，可以从数据包中获取 RSSI 值。用从数据包中得到的数据减去 110 (D) 就是 RSSI。例如，数据是 0x33, 0x33 减去 110 是-59。所以 RSSI 是-59。

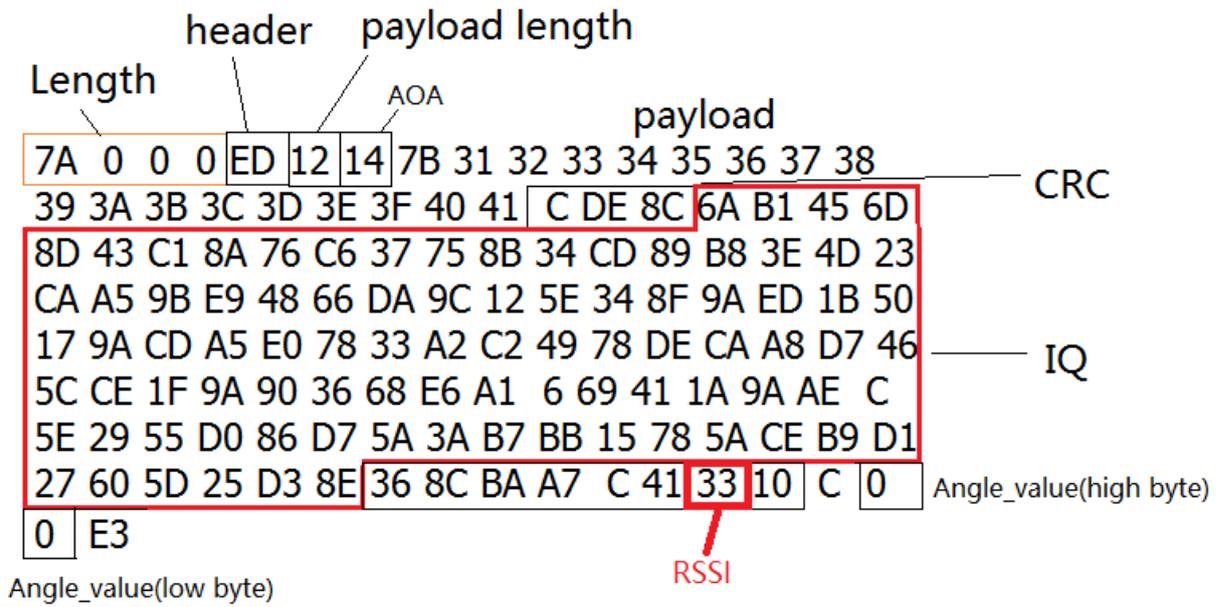


Figure 5.16: 三角板 (RSSI) 的数据包示例

6 现场测试结果

6.1 参考板 1 的现场测试结果

室内测试环境如图 6.1 所示。多天线的一端可以切换天线以发送或接收恒定单音，附着在距离地面 2 米高的墙壁上。在另一端，一个单天线 BLE 模块被放在距离多天线一端 5 米远的地方，以 5 度步长进行测试，测试结果如图 6.2 所示。



Figure 6.1: 参考板 1 的测试环境

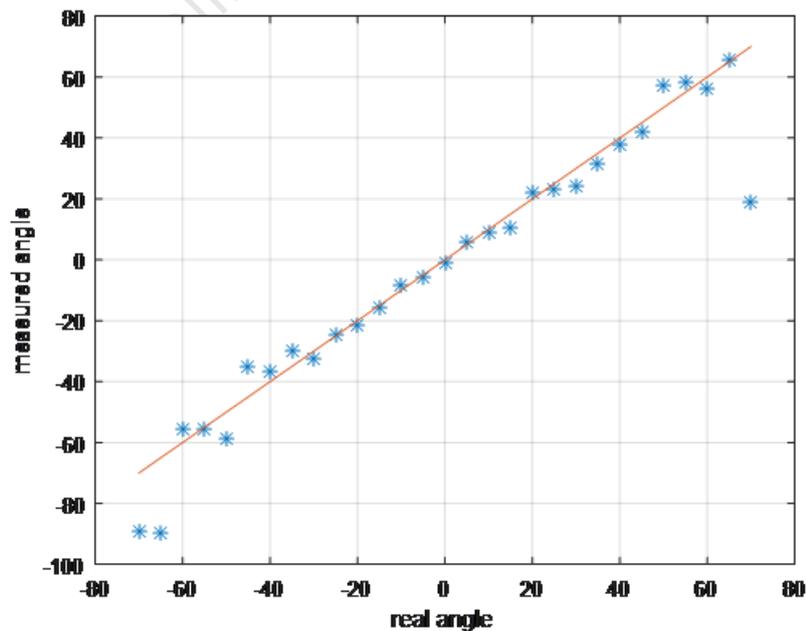


Figure 6.2: 5 米室内测试结果

下图是不同场景下的测试结果，表示误差绝对值的均值 vs 实际角度。参考板 1 放置的高度为 1 米或 2 米，单天线 BLE 模块放置的高度为 1 米。按照下面的测试，从墙到单天线适配器的距离分别为 0.5、1、2、3 和 5 米。

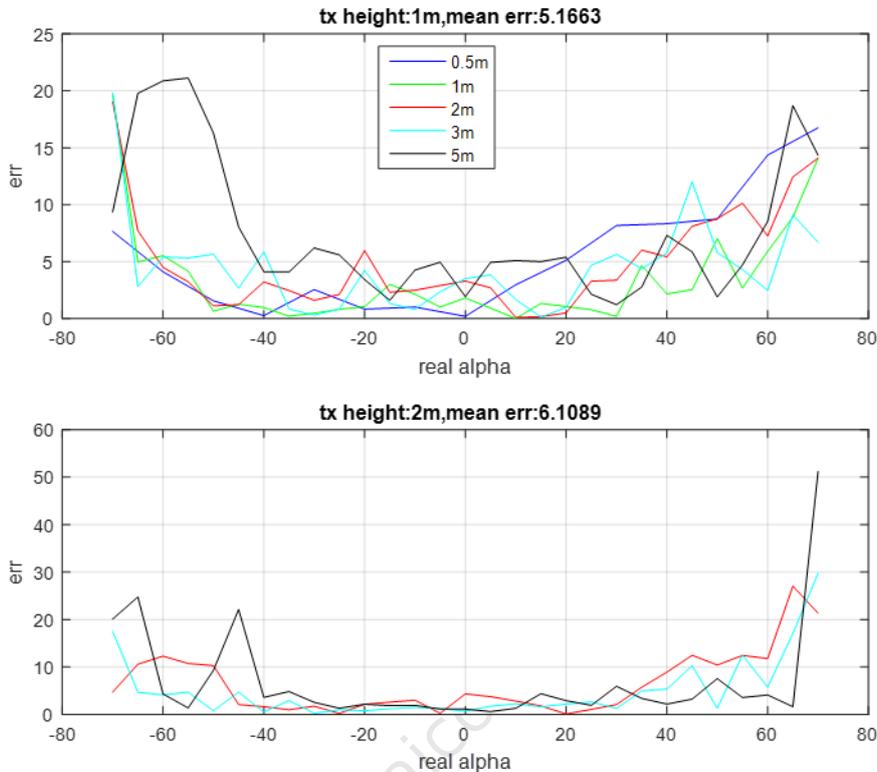


Figure 6.3: 误差绝对值均值 vs 实际角度

6.2 参考板 2 的现场测试结果

室内测试环境如图 6.4 所示。多天线的一端可以切换天线以发送或接收恒定频率扩展，挂在 1.8 米高的支架上。另一端是单天线 BLE 模块，放在地板上或 0.9 米高、距离支架 1 米或 2 米远的地方，将高度标记为 H，距离标记为 D。以 5 度步长进行测试，测试结果如图 6.5 和图 6.6 所示。

场景 1: $D = 1, H = 0$;

场景 2: $D = 2, H = 0$;

场景 3: $D = 1, H = 0.9$;

场景 4: $D = 2, H = 0.9$.

同时测量 α 和 θ 的角度。



Figure 6.4: 参考板 2 的测试环境

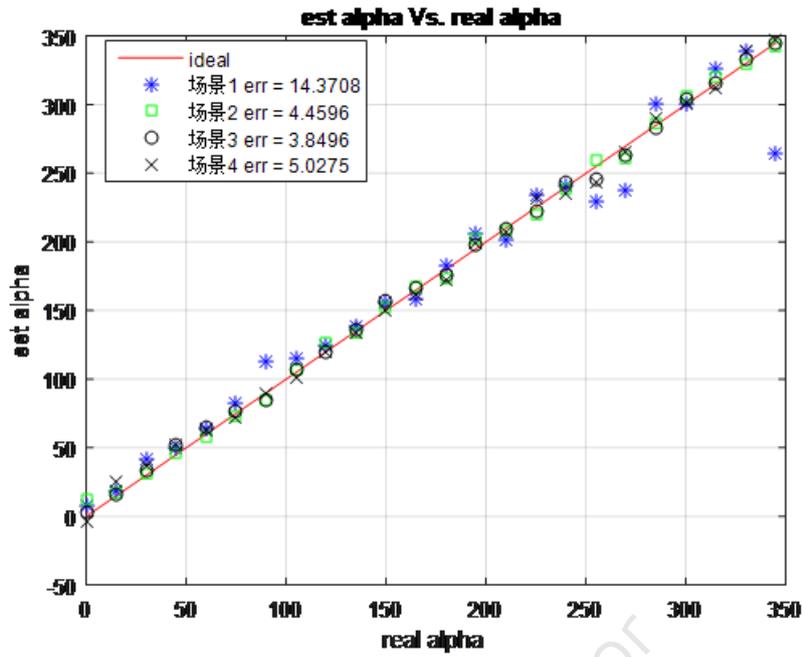


Figure 6.5: 估算 alpha vs 实际 alpha

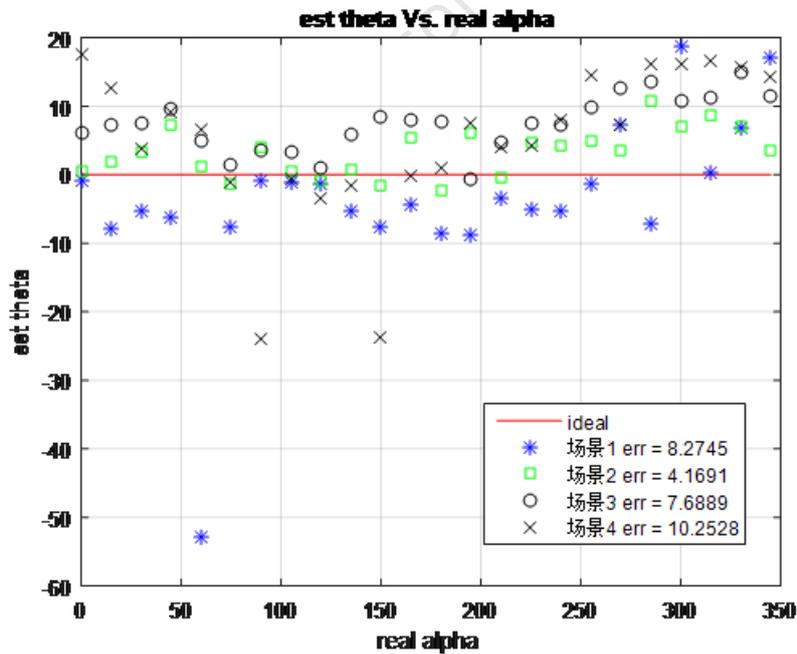


Figure 6.6: 估算 theta vs 实际 alpha

6.3 参考板 3 的现场测试结果

测试环境如图 6.7 所示。多天线的一端可以切换天线以发送或接收恒定频率扩展，挂在 2.5 米高的支架上。另一边是单天线 BLE 模块，放在地板上或 1.0 米高、距离支架 1 米、2 米或 3 米远的地方，将高度标记为 H，距

离标记为 D。以 5 度步长进行测试，测试结果如图 6.8 至 6.11 所示。

场景 1: $D = 1, H = 0$;

场景 2: $D = 2, H = 0$;

场景 3: $D = 3, H = 0$;

场景 4: $D = 1, H = 1$;

场景 5: $D = 2, H = 1$;

场景 6: $D = 3, H = 1$.

同时测量 α 和 θ 的角度。



Figure 6.7: 参考板 3 的测试环境

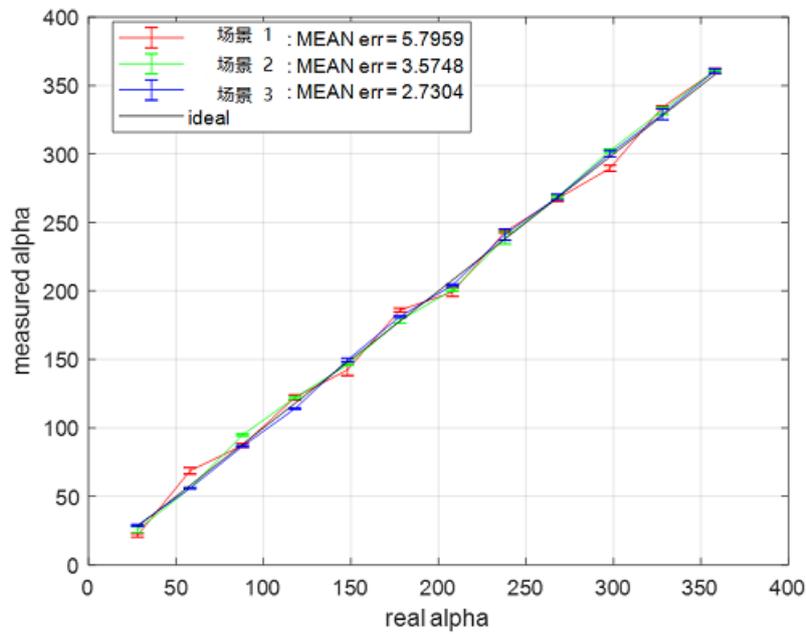


Figure 6.8: 场景 1、2、3 估算 alpha vs 实际 alpha

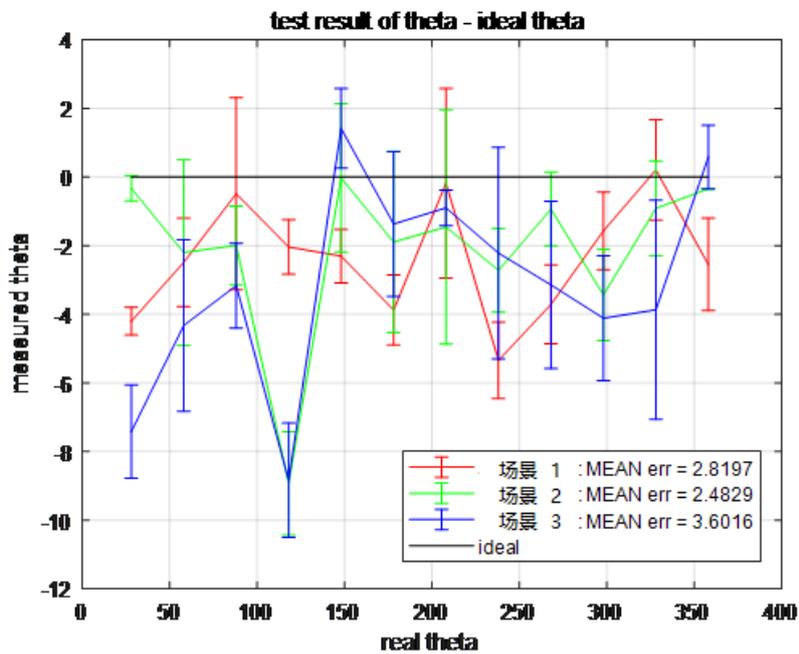


Figure 6.9: 场景 1、2、3 估算 theta vs 实际 alpha

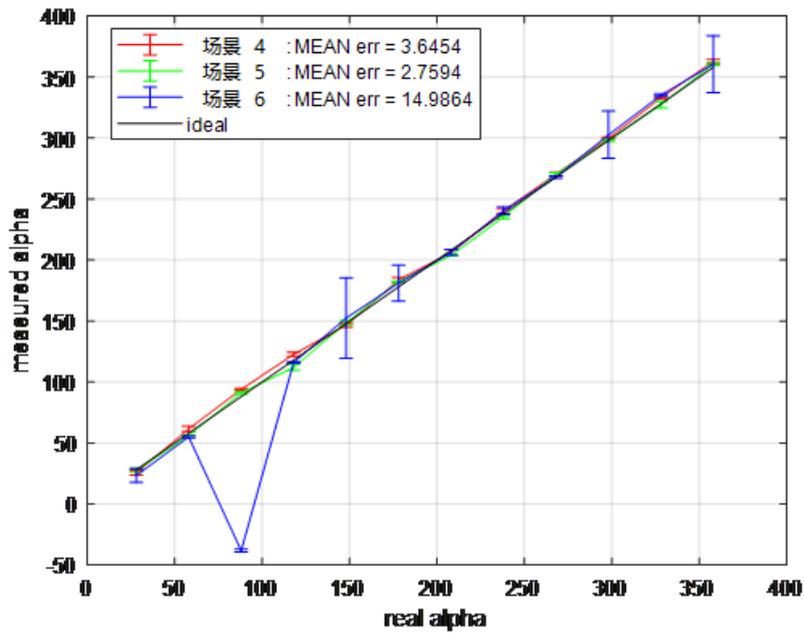


Figure 6.10: 场景 4、5、6 估算 alpha vs 实际 alpha

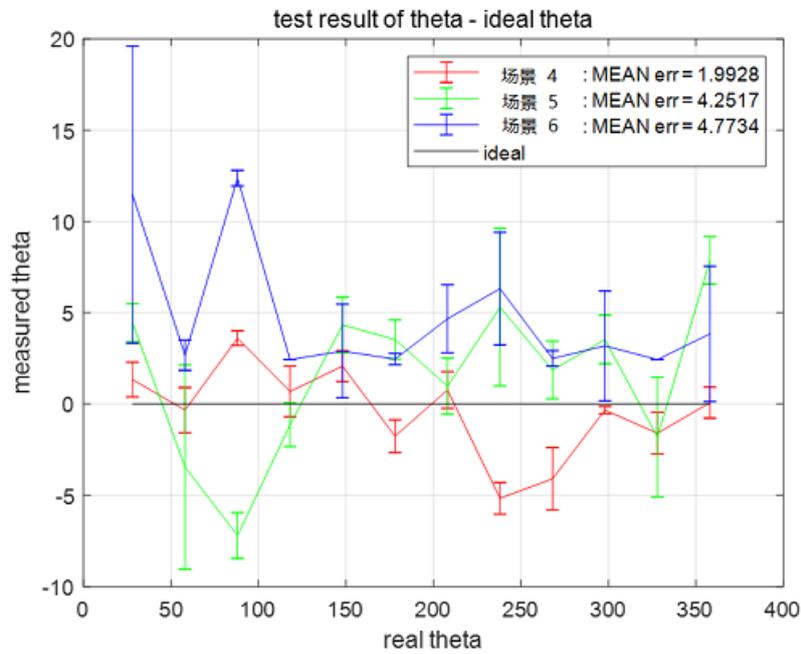


Figure 6.11: 场景 4、5、6 估算 theta vs 实际 alpha

7 参考板原理图

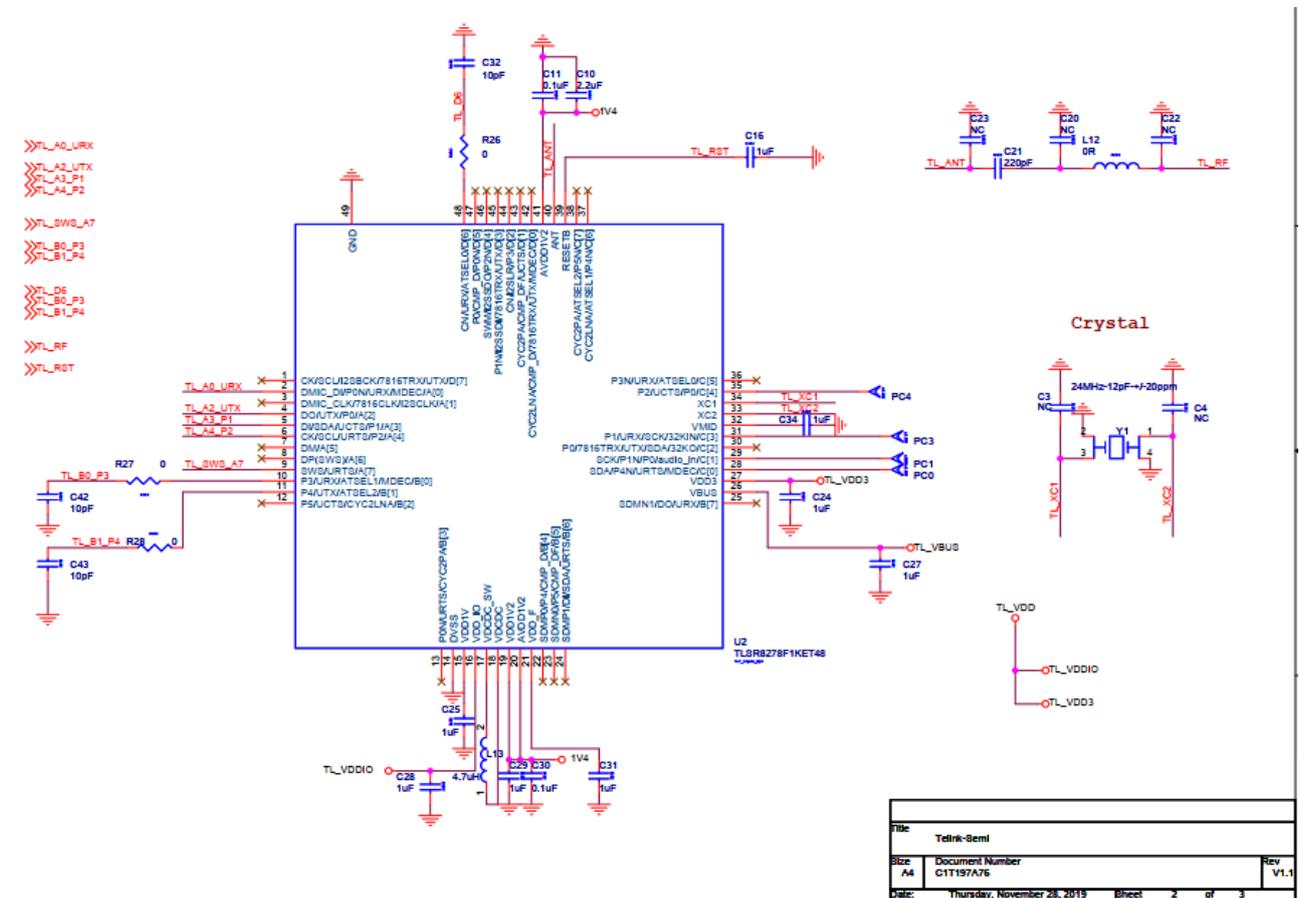


Figure 7.1: 参考板原理图第 1 部分

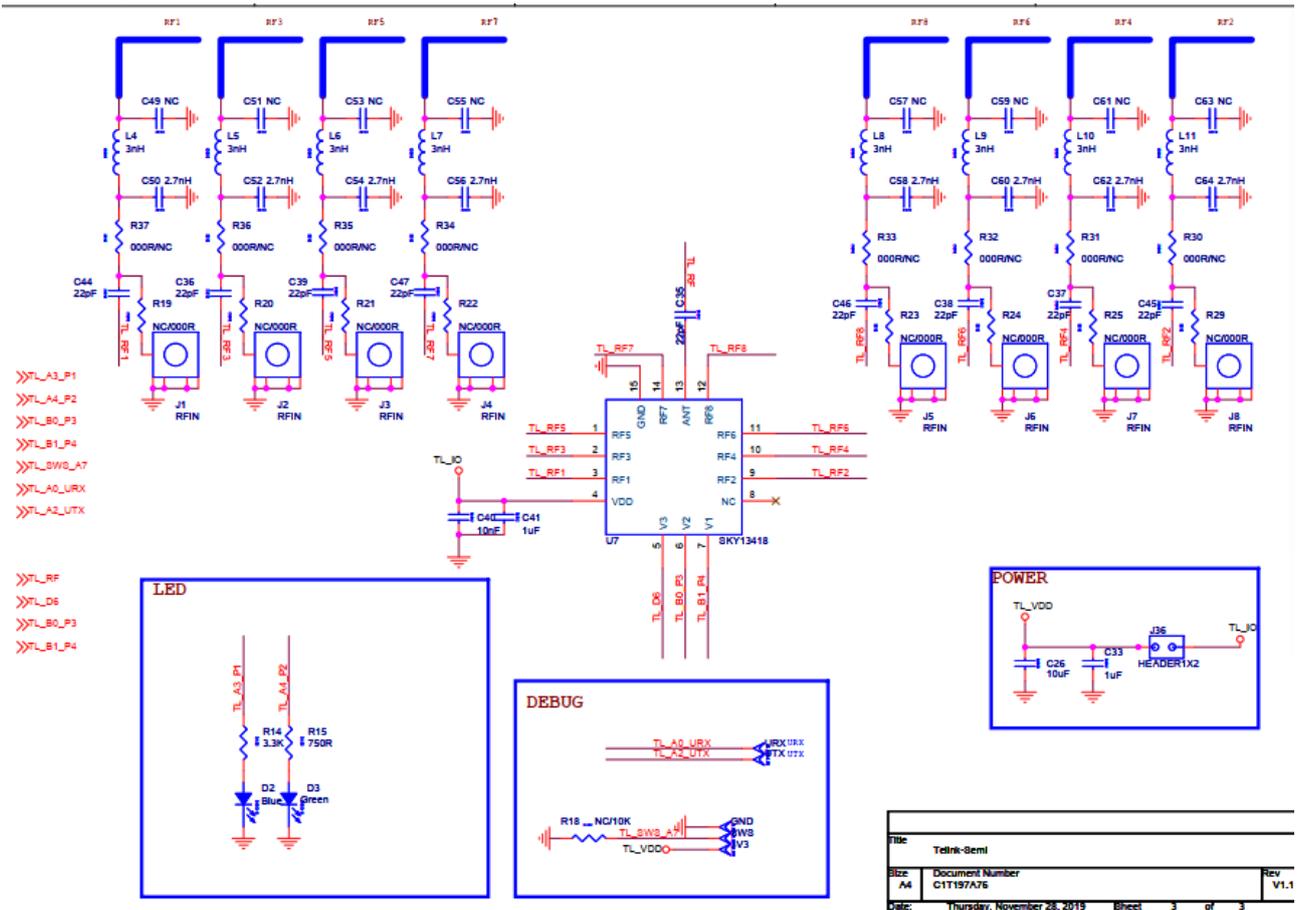


Figure 7.2: 参考板原理图第 2 部分