

APH001-UWB 模块 TWR 双边测距协议

文档更新时间：2018-06-1

1 定位帧基本结构

- 该定位帧遵循 IEEE 802.15.4 协议。IEEE 802.15.4 描述了低速率无线个人局域网的物理层和媒体接入控制协议。它属于 IEEE 802.15 工作组。IEEE 802.15.4 是 ZigBee, WirelessHART, MiWi, Thread 规范的基础。
- IEEE 802.15.4 MAC 层帧结构的设计是以用最低复杂度实现在多噪声无线信道环境下的可靠数据传输为目标的。每个 MAC 子层的帧都包含帧头、负载和帧尾三部分。帧头部分由帧控制信息、帧序列号和地址信息组成。MAC 子层的负载部分长度可变，负载的具体内容由帧类型决定。帧尾部分是帧头和负载数据的 16 位 CRC (FCS) 校验序列。
- 在 MAC 子层中设备地址有两种格式：16 位（两个字节）的短地址和 64 位（8 个字节）的扩展地址。16 位短地址是设备与个域网协调器关联时，由协调器分配的个域网内局部地址；64 位扩展地址则是全球唯一地址，在设备进入网络之前就分配好了。16 位短地址只能保证在个域网内部是唯一的，所以在使用 16 位短地址通信时需要结合 16 位的个域网网络标识符才有意义。两种地址类型地址信息的长度是不同的，所以 MAC 帧头的长度也是可变的。一个数据帧使用哪种地址类型由帧控制字段标识。

表 1 信标帧格式

2 字节	1 字节	2 字节	2 字节	2 字节	可变字节	2 字节
Frame Control (FC)	Sequence Number	PAN ID	Destination Address	Source Address	Ranging Message	FCS
MHR 帧头					MAC 负载	MFR 帧尾

1.1 Frame Control

Frame Control (FC)																
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Bit 9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	
1	0	0	0	0	0	1	0	0	0	DestAddrMode		0	0	SrcAddrMode		
Data Frame			SEC	PEND	ACK					0	1				0	1

1.2 Sequence Number

序列号，会自增。

1.3 PAN ID

PAN ID 是个域网地址。出现一般是伴随在确定信道以后的。PAN ID 其全称是 Personal Area Network ID，意即网络的 ID(即网络标识符)。固定值，设置为 0xDECA。

1.4 Destination Address

对方地址的 ID，关于地址，目前是由拨码开关所决定的，虽然 Mini3 开发板地址由 3bit 拨码开关决定，Mini3s / Mini3sPlus 地址由 AT 命令设置。通过以上方式设置地址的数量是有限制的，通过修改 STM32 源码可设置 65535 个不同的地址。

1.5 Source Address

本机地址。

1.6 Ranging Message

Ranging Message 有以下三种类型，在程序的不同阶段请对号入座。详细说明见第 2 章。

1.7 FCS

Frame Check Sequence (简称 FCS)，利用一定的校验方法，校验整个帧的数据。校验方法例如：如 CRC 校验、XOR 校验、LRC 校验等。这里的 FCS 是在操作一定寄存器下，由 DW1000 自动计算的。

2 Ranging Message

2.1 Poll Message

1 字节	1 字节
Function Code	Range number
0x81	-

这个是由 Tag 发起的一个 Poll message。

2.2 Response Message

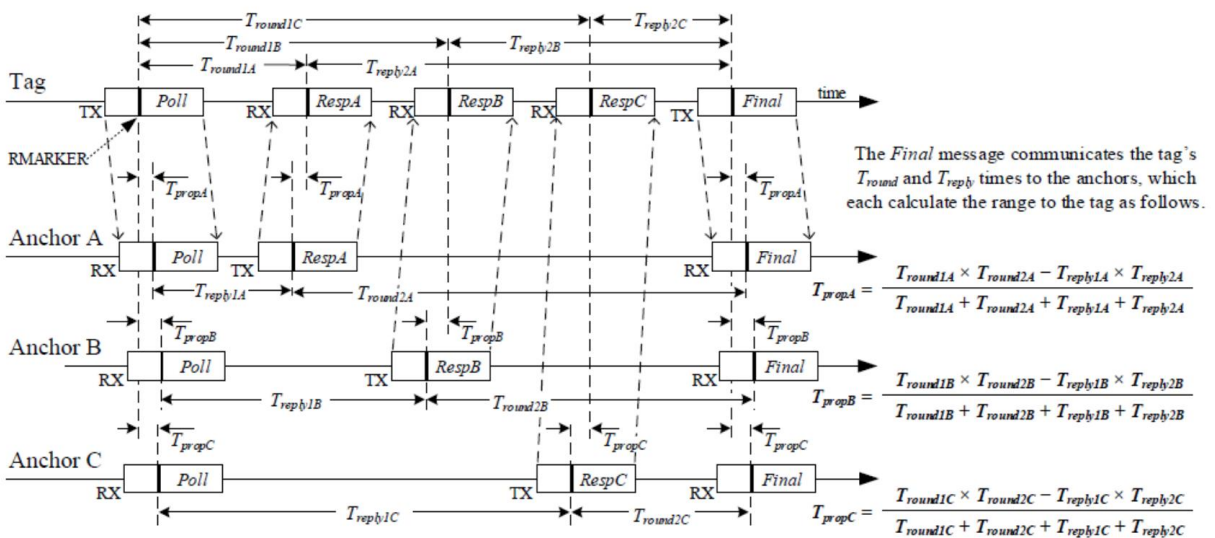
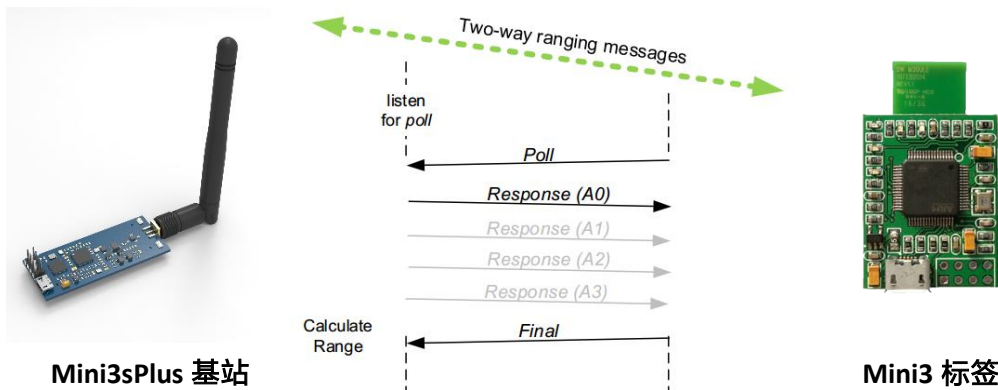
1 字节	2 字节	4 字节	1 字节
Function Code	Sleep Correction	ToF (n-1)	Range number
0x70	-	-	-

这个是由基站给标签发的，这里的 ToF 是上一个阶段的飞行时间。

2.3 Final Message

1 字节	1 字节	5 字节	5 字节	5 字节	5 字节	5 字节	5 字节	1 字节
Function Code	Range number	Poll TX time	A0 Resp RX time	A1 Resp RX time	A2 Resp RX time	A3 Resp RX time	Final TX time	Valid Resp
0x82	-	-	-	-	-	-	-	-

3 收发应答过程

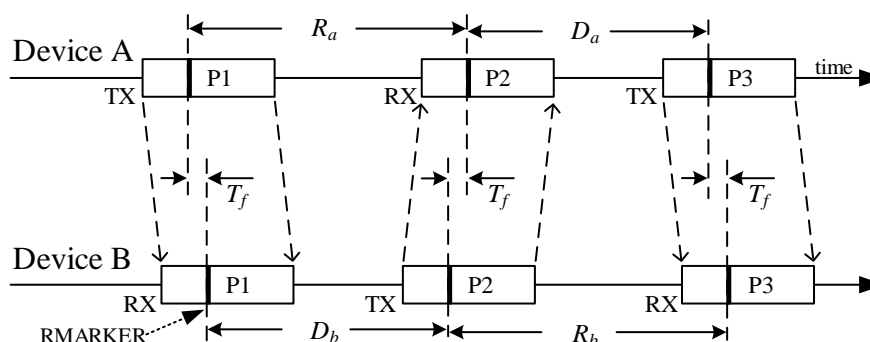


4 TOF 数学计算

```

1. // poll response round trip delay time is calculated as
2. // (anchorRespRxTime - tagPollTxTime) - (anchorRespTxTime - tagPollRxTime)
3. Ra = (int64)((anchorRespRxTime - tagPollTxTime) & MASK_40BIT);
4. Db = (int64)((inst->txu.anchorRespTxTime - inst->tagPollRxTime) & MASK_40BIT);
5.
6. // response final round trip delay time is calculated as
7. // (tagFinalRxTime - anchorRespTxTime) - (tagFinalTxTime - anchorRespRxTime)
8. Rb = (int64)((tagFinalRxTime - inst->txu.anchorRespTxTime) & MASK_40BIT);
9. Da = (int64)((tagFinalTxTime - anchorRespRxTime) & MASK_40BIT);
10.
11. RaRbxDaDb = (((double)Ra))*(((double)Rb)) - (((double)Da))*(((double)Db));
12.
13. RbyDb = ((double)Rb + (double)Db);
14. RayDa = ((double)Ra + (double)Da);
15.
16. tof = (int32) ( RaRbxDaDb /(RbyDb + RayDa) );

```



$$\hat{T}_{fab} = \frac{\hat{R}_a \hat{R}_b - \hat{D}_a \hat{D}_b}{\hat{R}_a + \hat{D}_a + \hat{R}_b + \hat{D}_b}$$

5 三边定位算法

有购买上位机源代码的客户，可在源文件中提取三边定位算法，使用 GetLocation 函数进行坐标计算，调用十分简单。经验证，该函数在普通嵌入式单片机上亦运行良好。其完整函数为：

```

1. int GetLocation(vec3d *best_solution, int use4thAnchor, vec3d* anchorArray, int *distanceArray)
2. {
3.     /* processing */
4. }

```

4 个参数，*best_solution 指向的是最终坐标的地址，vec3d 是一个结构体。第二个参数，如果使用 3 个基站进行定位，use4thAnchor 等于 0，如果使用 4 个基站进行定位，use4thAnchor 等于 1。anchorArray 需要传入的是基站的坐标位置。*distanceArray 是基站到标签的距离。

使用例子如下所示:

```
1. int main()
2. {
3.     int result = 0;
4.     vec3d anchorArray[4];
5.     vec3d report;
6.     int Range_deca[4];
7.     anchorArray[0].x = 0.000; //anchor0.x uint:m
8.     anchorArray[0].y = 0.000; //anchor0.y uint:m
9.     anchorArray[0].z = 2.000; //anchor0.z uint:m
10.
11.    anchorArray[1].x = -6.80; //anchor1.x uint:m
12.    anchorArray[1].y = 0.000; //anchor1.y uint:m
13.    anchorArray[1].z = 2.000; //anchor1.z uint:m
14.
15.    anchorArray[2].x = 0.000; //anchor2.x uint:m
16.    anchorArray[2].y = -10.8; //anchor2.y uint:m
17.    anchorArray[2].z = 2.000; //anchor2.z uint:m
18.
19.    anchorArray[3].x = 0.000; //anchor3.x uint:m
20.    anchorArray[3].y = -5.80; //anchor3.y uint:m
21.    anchorArray[3].z = 2.000; //anchor3.z uint:m
22.
23.    Range_deca[0] = 5784; //tag to A0 distance
24.    Range_deca[1] = 7021; //tag to A1 distance
25.    Range_deca[2] = 5995; //tag to A2 distance
26.    Range_deca[3] = 2000; //tag to A3 distance
27.
28.    result = GetLocation(&report, 0, &anchorArray[0], &Range_deca[0]);
29.
30.    printf("result = %d\r\n",result);
31.    printf("tag.x=%.3f\r\ntag.y=%.3f\r\ntag.z=%.3f\r\n",report.x,report.y,report.z);
32.    return 0;
33. }
```

所计算的结果, 保存在 report.x,report.y,report.z 中, 并通过 printf 函数打印出来。

6 参考文献

- [1] DecaRangeRTLS_ARM_Source_Code_Guide, Decawave
- [2] Indoor Positioning using Ultra-wideband Technology, Marcus Utter
- [3] IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)