trek1000 TWR ranging introduction

V1. 1

目 录

DISCLAIMER

YCHIOT has the right to update the product description without informing the customer. Changes in functions and specifications will be published in product errata or new versions of documents as much as possible. It is recommended that customers login to YCHIOT official website www.ychiot.com to download the latest product description documents.

LIFE SUPPORT POLICY

YCHIOT products are not authorized to be used in high safety areas (such as places where there is danger to life), because serious personal injury or death may be caused if the products are operated incorrectly. If a customer uses or sells YCHIOT products to high security areas, the customer needs to bear all the responsibilities; If this product is used in the field of high security, the customer needs to agree that YCHIOT and its agents are completely irresponsible.

Note! Electrostatic sensitive equipment. When using the product, take precautions to prevent permanent damage.

REGULATORY CERTIFICATION

All users who use this module for product development must obtain the approval of the local radio supervision and management department before marketing or selling the product, and the customer must assume all responsibilities for obtaining the approval from the relevant authorities.

1 Basic structure of positioning frame

The location frame follows IEEE 802.15.4 protocol. IEEE 802.15.4 describes the physical layer and media access control protocol of low-rate wireless personal network. It belongs to the IEEE 802.15 Working Group. IEEE 802.15.4 is the basis of ZigBee, WirelessHART, MiWi and Thread specifications.

The frame structure of IEEE 802.15.4 MAC layer is designed to achieve reliable data transmission in a multi-noise wireless channel environment with minimum complexity. The frame of each MAC sublayer contains three parts: frame header, load and frame tail. The frame header is composed of frame control information, frame sequence number and address information. The length of the load part of the MAC sublayer is variable, and the specific content of the load is determined by the frame type. The end of frame is a 16-bit CRC (FCS) check sequence of frame header and load data.

In the MAC sublayer, the device address has two formats: 16-bit (two bytes) short address and 64-bit (eight bytes) extended address. The 16-bit short address is the local address in the domain network assigned by the coordinator when the device is associated with the domain network coordinator; The 64-bit extended address is the only address in the world, and is assigned before the device enters the network. The 16-bit short address can only be guaranteed to be unique within the individual domain networks, so it is meaningful to combine the 16-bit individual domain network identifier when using the 16-bit short address for communication. The length of the address information of the two address types is different, so the length of the MAC frame header is also variable. The address type used by a data frame is identified by the frame control field.

Figure 1 frame format

Frame Control (FC)	Sequence Number	PAN ID	Destination Address	Source Address	Ranging Message	FCS
2 Bytes	1 Byte	2 Bytes	2 Bytes	2 Bytes	Variable byte	2 Bytes
MHR					MAC	MFR

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

The serial number will increase automatically.

1.3 PAN ID

PAN ID is a domain address. The occurrence usually follows the determination of the channel. The full name of PAN ID is Personal Area Network ID, which means the network ID (i.e. network identifier). Fixed value, set to 0xDECA.

1.4 Destination Address

Remote address.

1.5 Source Address

Local address.

1.6 Ranging Message

There are three types of Ranging Message. Please take the right seat at different stages of the program. See Chapter 2 for details.

1.7 FCS

Frame Check Sequence (FCS) uses a certain verification method to verify the data of the entire frame. Verification methods such as CRC verification, XOR verification, LRC verification, etc. The FCS here is automatically calculated by DW1000 under the operation of certain registers.

2 Ranging Message

2.1 Poll Message

Function Code	Range number
1byte	1 byte
0x81	-

This is a poll message initiated by the tag.

2.2 Response Message

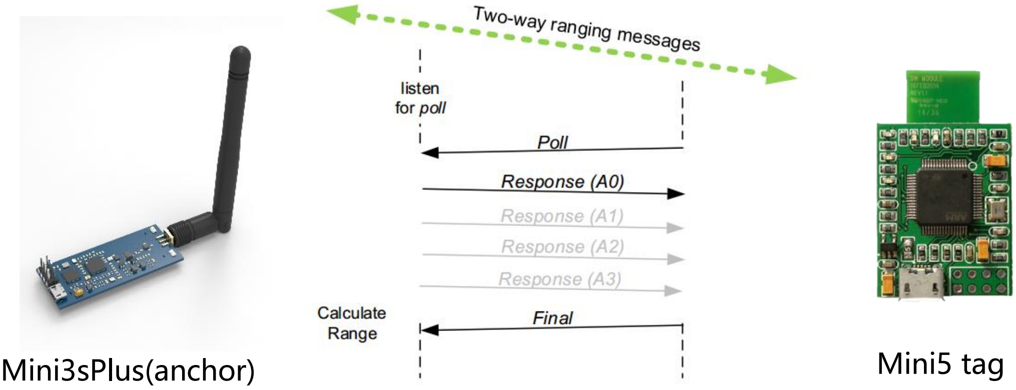
Function Code	Sleep Correction	ToF (n-1)	Range number
1 byte	2 bytes	4 bytes	1 byte
0x70	-	-	-

This is the data sent by the anchor to the tag. The ToF here is the flight time of the previous frame.

2.3 Final Message

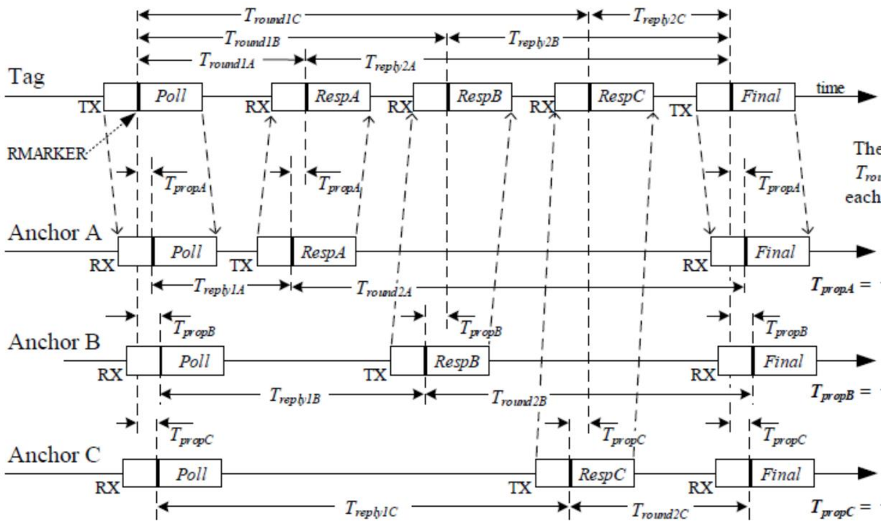
Function Code	Range number	Poll TX time	A0 Resp RX time	A1 Resp RX time	A2 Resp RX time	A3 Resp RX time	Final TX time	Valid Resp
1 byte	1 byte	5 bytes	5 bytes	5 bytes	5 bytes	5 bytes	5 bytes	1 byte
0x82	-	-	-	-	-	-	-	-

3 Communication process



Mini3sPlus(anchor)

Mini5 tag



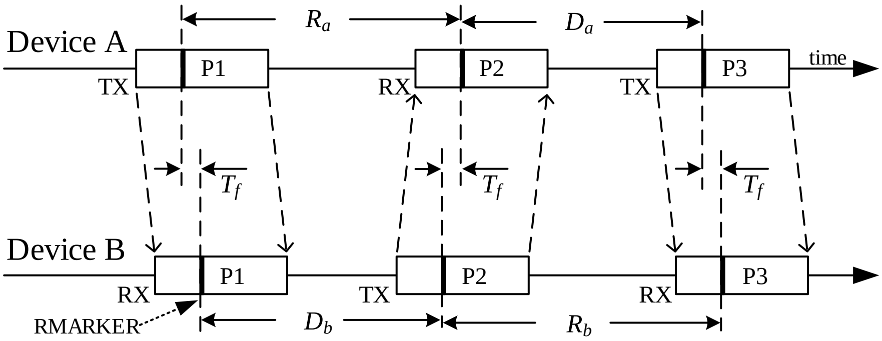
$$T_{propA} = \frac{T_{round1A} \times T_{round2A} - T_{reply1A} \times T_{reply2A}}{T_{round1A} + T_{round2A} + T_{reply1A} + T_{reply2A}}$$

$$T_{propB} = \frac{T_{round1B} \times T_{round2B} - T_{reply1B} \times T_{reply2B}}{T_{round1B} + T_{round2B} + T_{reply1B} + T_{reply2B}}$$

$$T_{propC} = \frac{T_{round1C} \times T_{round2C} - T_{reply1C} \times T_{reply2C}}{T_{round1C} + T_{round2C} + T_{reply1C} + T_{reply2C}}$$

4 TOF Mathematical calculation

4.1 TOF flow chart



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

4.2 Relevant code

```
// poll response round trip delay time is calculated as
// (anchorRespRxTime - tagPollTxTime) - (anchorRespTxTime - tagPollRxTime)
Ra = (int64)((anchorRespRxTime - tagPollTxTime) & MASK_40BIT);
Db = (int64)((inst->txu.anchorRespTxTime - inst->tagPollRxTime) & MASK_40BIT);

// response final round trip delay time is calculated as
// (tagFinalRxTime - anchorRespTxTime) - (tagFinalTxTime - anchorRespRxTime)
Rb = (int64)((tagFinalRxTime - inst->txu.anchorRespTxTime) & MASK_40BIT);
Da = (int64)((tagFinalTxTime - anchorRespRxTime) & MASK_40BIT);

RaRbxDaDb = (((double)Ra))*(((double)Rb))  - (((double)Da))*(((double)Db));

RbyDb = ((double)Rb + (double)Db);
RayDa = ((double)Ra + (double)Da);

tof = (int32) ( RaRbxDaDb /(RbyDb + RayDa) );
```

5 Trilateration algorithm

5.1 Locating function call methods

Customers who have purchased the source code of the upper computer can extract the trilateral location algorithm from the source file and use the GetLocation function to calculate the coordinates, which is very simple to call. It is verified that this function also works well on ordinary embedded single-chip microcomputer. Its complete function is,

```
int GetLocation(vec3d *best_solution, int use4thAnchor, vec3d* anchorArray, int
*distanceArray)
{
    /* processing */
}
```

4 parameters, * best_ Solution points to the address of the final coordinate, and vec3d is a structure. The second parameter is that if three base stations are used for positioning, use4thAnchor is equal to 0. If four base stations are used for positioning, use4thAnchor is equal to 1. AnchorArray needs to pass in the coordinate position of the base station* DistanceArray is the distance from the base station to the tag. Examples are as follows:

```
int main()
{
    int result = 0;
    vec3d anchorArray[4];
    vec3d report;
    int Range_deca[4];
    anchorArray[0].x = 0.000; //anchor0.x uint:m
    anchorArray[0].y = 0.000; //anchor0.y uint:m
    anchorArray[0].z = 2.000; //anchor0.z uint:m

    anchorArray[1].x = -6.80; //anchor1.x uint:m
    anchorArray[1].y = 0.000; //anchor1.y uint:m
    anchorArray[1].z = 2.000; //anchor1.z uint:m

    anchorArray[2].x = 0.000; //anchor2.x uint:m
    anchorArray[2].y = -10.8; //anchor2.y uint:m
    anchorArray[2].z = 2.000; //anchor2.z uint:m
```

```
    anchorArray[3].x = 0.000; //anchor3.x uint:m
    anchorArray[3].y = -5.80; //anchor3.y uint:m
    anchorArray[3].z = 2.000; //anchor3.z uint:m

    Range_deca[0] = 5784; //tag to A0 distance
    Range_deca[1] = 7021; //tag to A1 distance
    Range_deca[2] = 5995; //tag to A2 distance
    Range_deca[3] = 2000; //tag to A3 distance

     result = GetLocation(&report, 0, &anchorArray[0], &Range_deca[0]);

    printf("result = %d\r\n",result);
    printf("tag.x=%.3f\r\ntag.y=%.3f\r\ntag.z=%.3f\r\n",report.x,report.y,report.z);

    return 0;
}
```

The calculated results are saved in report. x, report. y, and report. z, and printed through the printf function.

6 Document Management Information Sheet

Subject	YCHIOT UWB multi-anchor positioning solutions
Version	V1.1
Reference documents	[1] DecaRangeRTLS_ARM_Source_Code_Guide, Qorvo [2] Indoor Positioning using Ultra-wideband Technology, Marcus Utter [3] IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)
Creation time	2018/06/01
Founder	Lynn
Latest release date	2023/01/01

Modifier	Date	Document change history
Lynn	2018-06-01	V1.0 Release of V1.0 documentation
Lynn	2023-01-01	V1.1 Change to YCHIOT new document style