|  |  |
| --- | --- |
| 通信接口 | UART |
| 默认波特率 | 115200 |
| 数据位 | 8 |
| 停止位 | 1 |
| 奇偶校验 | None |

标准数据输出格式（默认）：  
数据结构：每个数据包为 10Byte。包含数据类型（Type）、数据（data）数据校验字节（CheckSum）等。数据格式为 16 进制（HEX）。具体数据编码详见表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Byte0-1 | Byte2 | ByteType | ByteData | ByteType | ByteData | …… | Bytelast |
| 0x49  0x39 | Type\_Num | Type | Date | Type | Data | …… | 0x55  0x39 |

注：杭州电子科技大学hzdzkjdx九键为49395539

|  |  |
| --- | --- |
| 数据编码解释 | |
| Byte0 | 0x49，每一帧都相同 |
| Byte1 | 0x39，每一帧都相同 |
| Byte2 | Type\_Num帧状态，前4位表示数据种类，后4位为数据个数，例：0x15表示数据类型为①，发送五个数据  数据种类分四类：①int16\_t②int32\_t③uint16\_t④uint32\_t |
| ByteType | 数据类型，八位 |
| ByteData | Data,若数据类型为1、3则为16位，若数据类型为2、4则为32位 |
| Bytelast2 | 0x55，每一帧都相同 |
| Bytelast | 0x39，每一帧都相同 |

/\*\*\*\*\*\*\*\*\*\*\*connect.h\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef \_connect\_h

#define \_connect\_h

#include "include.h"

typedef struct

{

uint8\_t Type;

uint8\_t Date\_1;

uint8\_t Date\_2;

uint8\_t Date\_3;

uint8\_t Date\_4;

}Connect\_TypedefU;

typedef struct

{

uint8\_t Type;

int8\_t Date\_1;

int8\_t Date\_2;

int8\_t Date\_3;

int8\_t Date\_4;

}Connect\_Typedef;

enum

{

SVEL=1,

SREMOTE,

SLAS

};

extern Connect\_TypedefU unsigned\_connet;

extern Connect\_Typedef signed\_connet;

extern uint8\_t SEND\_STA;

void Set\_TypeU(Connect\_TypedefU \*co,uint8\_t type);

void Set\_Type(Connect\_Typedef \*co,uint8\_t type);

void Set\_Datau32(Connect\_TypedefU \*co,uint32\_t data);

void Set\_Datau16(Connect\_TypedefU \*co,uint16\_t data);

void Set\_Data32(Connect\_Typedef \*co,int32\_t data);

void Set\_Data16(Connect\_Typedef \*co,int16\_t data);

void Send\_Begin(uint8\_t STA);

void Send\_End(void);

void Send\_MessageU16(Connect\_TypedefU \*co);

void Send\_Message16(Connect\_Typedef \*co);

void Send\_MessageU32(Connect\_TypedefU \*co);

void Send\_Message32(Connect\_Typedef \*co);

void publish\_vel(void);

void publish\_remote(void);

void publish\_las(void);

#define MSG\_vel\_x 0X01

#define MSG\_vel\_y 0X02

#define MSG\_vel\_z 0X03

#define MSG\_distance\_f 0X04 //前距离

#define MSG\_distance\_b 0X05 //后距离

#define MSG\_distance\_l 0X06 //左距离

#define MSG\_distance\_r 0X07 //右距离

#define MSG\_remote\_ch0 0x08 //遥控器ch0

#define MSG\_remote\_ch1 0x08 //遥控器ch1

#define MSG\_remote\_ch2 0x09 //遥控器ch2

#define MSG\_remote\_ch3 0x0A //遥控器ch3

#define MSG\_remote\_s1 0x0B //遥控器s1

#define MSG\_remote\_s2 0x0C //遥控器s2

#endif

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*connect.c\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "connect.h"

Connect\_TypedefU unsigned\_connet;

Connect\_Typedef signed\_connet;

uint8\_t SEND\_STA;

void Set\_TypeU(Connect\_TypedefU \*co,uint8\_t type)

{

co->Type=type;

}

void Set\_Type(Connect\_Typedef \*co,uint8\_t type)

{

co->Type=type;

}

void Set\_Datau32(Connect\_TypedefU \*co,uint32\_t data)

{

uint8\_t data1=(uint8\_t)data&0x000000FF;

uint8\_t data2=(uint8\_t)(data>>8)&0x000000FF;

uint8\_t data3=(uint8\_t)(data>>16)&0x000000FF;

uint8\_t data4=(uint8\_t)(data>>24)&0x000000FF;

co->Date\_1=data1;

co->Date\_2=data2;

co->Date\_3=data3;

co->Date\_4=data4;

}

void Set\_Datau16(Connect\_TypedefU \*co,uint16\_t data)

{

uint8\_t data1=(uint8\_t)data&0x00FF;

uint8\_t data2=(uint8\_t)(data>>8)&0x00FF;

co->Date\_1=data1;

co->Date\_2=data2;

}

void Set\_Data32(Connect\_Typedef \*co,int32\_t data)

{

int8\_t data1=(int8\_t)data&0x000000FF;

int8\_t data2=(int8\_t)(data>>8)&0x000000FF;

int8\_t data3=(int8\_t)(data>>16)&0x000000FF;

int8\_t data4=(int8\_t)(data>>24)&0x000000FF;

co->Date\_1=data1;

co->Date\_2=data2;

co->Date\_3=data3;

co->Date\_4=data4;

}

void Set\_Data16(Connect\_Typedef \*co,int16\_t data)

{

int8\_t data1=(int8\_t)data&0x00FF;

int8\_t data2=(int8\_t)(data>>8)&0x00FF;

co->Date\_1=data1;

co->Date\_2=data2;

}

void Send\_MessageU16(Connect\_TypedefU \*co)

{

usart1\_write(co->Type);

usart1\_write(co->Date\_1);

usart1\_write(co->Date\_2);

}

void Send\_MessageU32(Connect\_TypedefU \*co)

{

usart1\_write(co->Type);

usart1\_write(co->Date\_1);

usart1\_write(co->Date\_2);

usart1\_write(co->Date\_3);

usart1\_write(co->Date\_4);

}

void Send\_Message16(Connect\_Typedef \*co)

{

usart1\_write(co->Type);

usart1\_write(co->Date\_1);

usart1\_write(co->Date\_2);

}

void Send\_Message32(Connect\_Typedef \*co)

{

usart1\_write(co->Type);

usart1\_write(co->Date\_1);

usart1\_write(co->Date\_2);

usart1\_write(co->Date\_3);

usart1\_write(co->Date\_4);

}

void Send\_Begin(uint8\_t STA)

{

usart1\_write(0x49);

usart1\_write(0x39);

switch(STA)

{

case SVEL:

usart1\_write(0x13); SEND\_STA=True;

break;

case SREMOTE:

usart1\_write(0x36); SEND\_STA=True;

break;

case SLAS:

usart1\_write(0x34); SEND\_STA=True;

break;

default:

SEND\_STA=False;break;

}

}

void Send\_End(void)

{

usart1\_write(0x55);

usart1\_write(0x39);

SEND\_STA=False;

}

void publish\_vel(void)

{

Send\_Begin(SVEL);

if(SEND\_STA)

{

Set\_Type(&signed\_connet,MSG\_vel\_x);

Set\_Data16(&signed\_connet,(int16\_t)required\_linear\_vel\_x\*100);

Send\_Message16(&signed\_connet);

Set\_Type(&signed\_connet,MSG\_vel\_y);

Set\_Data16(&signed\_connet,(int16\_t)required\_linear\_vel\_y\*100);

Send\_Message16(&signed\_connet);

Set\_Type(&signed\_connet,MSG\_vel\_z);

Set\_Data16(&signed\_connet,(int16\_t)required\_angular\_vel\*100);

Send\_Message16(&signed\_connet);

}

Send\_End();

}

void publish\_remote(void)

{

Send\_Begin(SREMOTE);

if(SEND\_STA)

{

Set\_TypeU(&unsigned\_connet,MSG\_remote\_ch0);

Set\_Datau16(&unsigned\_connet,(uint16\_t)RC\_CtrlData.ch0);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_remote\_ch1);

Set\_Datau16(&unsigned\_connet,(uint16\_t)RC\_CtrlData.ch1);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_remote\_ch2);

Set\_Datau16(&unsigned\_connet,(uint16\_t)RC\_CtrlData.ch2);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_remote\_ch3);

Set\_Datau16(&unsigned\_connet,(uint16\_t)RC\_CtrlData.ch3);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_remote\_s1);

Set\_Datau16(&unsigned\_connet,(uint16\_t)RC\_CtrlData.s1);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_remote\_s2);

Set\_Datau16(&unsigned\_connet,(uint16\_t)RC\_CtrlData.s2);

Send\_MessageU16(&unsigned\_connet);

}

Send\_End();

}

void publish\_las(void)

{

Send\_Begin(SVEL);

if(SEND\_STA)

{

Set\_TypeU(&unsigned\_connet,MSG\_distance\_f);

Set\_Datau16(&unsigned\_connet,(uint16\_t)Distance\_F);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_distance\_b);

Set\_Datau16(&unsigned\_connet,(uint16\_t)Distance\_B);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_distance\_l);

Set\_Datau16(&unsigned\_connet,(uint16\_t)Distance\_L);

Send\_MessageU16(&unsigned\_connet);

Set\_TypeU(&unsigned\_connet,MSG\_distance\_r);

Set\_Datau16(&unsigned\_connet,(uint16\_t)Distance\_R);

Send\_MessageU16(&unsigned\_connet);

}

Send\_End();

}