#define WOOZOOM\_POWER\_DATA\_TYPE\_SIGNATURE                     0xdf8b928d07f48607

#define WOOZOOM\_POWER\_DATA\_TYPE\_ID                   5201

1. 通信包的信号和数据ID

void register\_callback(uint16\_t data\_type\_id, CanardOnTransferReception callback);

//注册ID, 和callback函数

//注册数据ID和回调函数的绑定

static uint8\_t registered\_callback = 0; //数量

static uint16\_t uavcan\_callback\_types[MAX\_SUPPORTED\_CALLBACK]; //ID

static CanardOnTransferReception uavcan\_callbacks[MAX\_SUPPORTED\_CALLBACK];//函数

register\_callback(WOOZOOM\_LED\_VALUE\_DATA\_TYPE\_ID, handleLedValueMessage);

//注册下面2个ID

#define WOOZOOM\_LED\_VALUE\_DATA\_TYPE\_ID 5101

*void* handleLedValueMessage(CanardInstance\* ins, CanardRxTransfer\* transfer)

void canardInit(CanardInstance\* out\_ins,

void\* mem\_arena,

size\_t mem\_arena\_size,

CanardOnTransferReception on\_reception,

CanardShouldAcceptTransfer should\_accept,

void\* user\_reference)

{

out\_ins->on\_reception = on\_reception;

out\_ins->should\_accept = should\_accept;

定义回调函数

canardInit(&canard, canard\_memory\_pool, sizeof(canard\_memory\_pool), onTransferReceived, shouldAcceptTransfer, NULL);

//判断

if (ins->should\_accept(ins, &data\_type\_signature, data\_type\_id, transfer\_type, source\_node\_id))

//接收完数据后的处理

ins->on\_reception(ins, &rx\_transfer);

//执行我们的回调函数

static void onTransferReceived(CanardInstance\* ins,

CanardRxTransfer\* transfer) if (idx<registered\_callback) {

uavcan\_callbacks[idx](ins, transfer);

//通过数据ID返回信号

bool ledMessageAccept(const CanardInstance\* ins,

-》if (data\_type\_id == WOOZOOM\_LED\_VALUE\_DATA\_TYPE\_ID) {

\*out\_data\_type\_signature = WOOZOOM\_LED\_VALUE\_DATA\_TYPE\_SIGNATURE;

//执行回调函数

void handleLedValueMessage(CanardInstance\* ins, CanardRxTransfer\* transfer)