DLSx0 Data Repeater User Manual

(Document version V1.0.0)

- UART←→UART/LoRA/LTE
- LoRA←→LoRA/UART/LTE
- LTE←→UART/LoRA
- Repeater, Gateway
- Data Recorder
- Wireless Sensor

HEBEI WINCOM TECHNOLOGY CO.,LTD

2021-07

目录

TEL: 0316-3093523 010-61591202 Email: INFO@GEO-INS.COM

OVERVIEW

Thank you for choosing our product!

DLSxx is a low-power data repeater with built-in battery and LoRA and LTE (4G) wireless. It can realize transparent data transmission, repeater and gateway functions through preset protocols.Using the technology of "sleep + wireless wake up", it can work autonomously for a long time under the condition of built-in battery.With the built-in data storage function, the received data can be stored in the Flash for reading and viewing at any time.It has the functions of real-time data forwarding and scheduled centralized data forwarding.Open all communication parameters, can be integrated with other manufacturers of similar products.

Note: This manual applies to DLSx0_xxxx (see "Product Selection" for details).

APPLICATION

- ◆ Automatic measurement and control LoRA-4G workstation
- ◆ Serial interface device to wireless
- ◆ LoRA repeater and frequency changer
- ◆ Match different LoRAs
- ◆ Wireless data recorder
- ◆ LoRA – 4G gateway

PRODUCT SELECTION

Model: DLS-AB_xUXX

- ■ DLS: Product type identifier, fixed as DLS (Double L System), L is LoRA and LTE.
- ■ A: The number of built-in LoRA, can be 0, 1, or 2.
- ■ B: The number of built-in LTE, can be 0, 1, or 2.
- ■ xUXX: UART port definition code.Example: 2UT2 indicates two UART interfaces, TTL and RS232.
  - ◆ xU: The number of UART interfaces, can be 1 or 2.
  - ◆ XX: UART port type. 2 – RS232, 4 – RS485, and T – TTL.

Recommended model

Model	Number of digital interface			Sensor interface				Built-in Battery
	UART	LoRA	LTE					
DLS10_1U2	1	1	0					1800mAH*3
DLS10_1U4	1	1	0					1800mAH*3
DLS10_2U24	2	1	0					1800mAH*3
DLS20_1U2	1	2	0					1800mAH*3
DLS11_1U2	1							1800mAH*3
DLS11_1U4	1	1	1					1800mAH*3
This table does not list all models Internal battery refers to the maximum capacity of the internal battery that can be installed.A built-in battery is not a standard option.								

PARAMETERS/SPECIFICATIONS

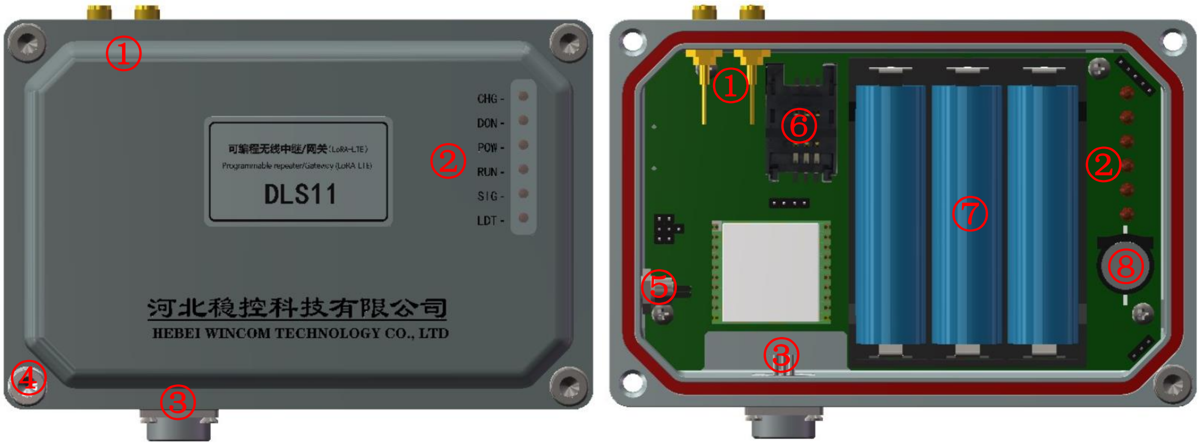
Test conditions and environment: Unless otherwise specified, the following indicators are measured at room temperature of 25℃.

Object	Conditions	Value			Unit
		MIN	STD	MAX	
Size	148x98x43 (BxCxD)				mm^3
IP Grade	IP65				
Power	1~3 18650 lithium + external charging interface				
Battery capacity	3*2500mAH				
ExPower		5.5	12	24	V
Power consumption (Battery powered)	IDLE State		3.3		mA
	STOP State		8		uA
	Real-time Receive		17.5		mA
	LoRA Send 100mW	25	115	125	mA
	LoRA Send 500mW				
	Transient peak		1.2		A
Battery life①	Built-in battery (3) Forwarded every minute		3		Years
Temperature	Operating temperature	-20		80	℃
	Storage temperature	-60		120	
Flash		4			MByte
		16384			
LoRA	433MHz	420		450	MHz
	868MHz	854		884	MHz
	number of channels		15		
	Air rate	300	2604	37500	bps
LTE	LTE-TDD B38/B39/B40/B41 LTE-FDDB1/B3/B8 TD-SCDMA B34/B39 UMTS/HSDPA/HSPA+ B1/B8 CDMA 1X/EVDO BC0 GSM/GPRS/EDGE900/1800 MHz				
UART Baud rate		1200	115200	460800	bps

Note ① : Power loss caused by battery self-discharge and frequent wake up in complex networks is not considered.

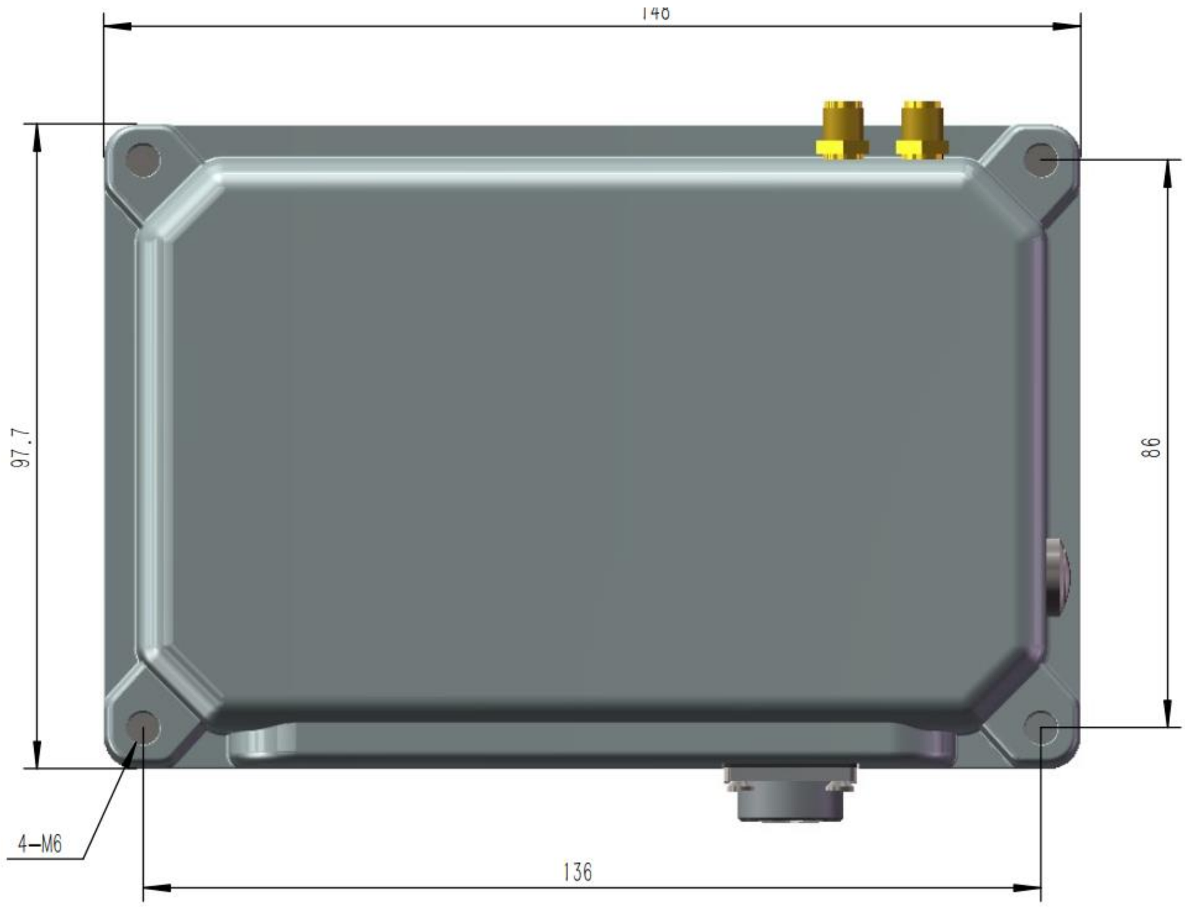
STRUCTURE/MOUNTING DIMENSIONS

Structure



① antenna	②Indicator leds	③ Power&Communication	④Fixing screw
⑤ Test key	⑥SIM Socked	⑦ Compartment	⑧Button battery

Mounting Dimensions



Mounting Dimensions（Bottom）

INDICATOR LEDs

Indicator Function

Symbol	Name	Status	Description	Node
CHG	Charging	ON	Be Charging…	
DON	Charging done	ON	Charging completed	
POW	Power	ON	Be Working…	Only DLSx1(LTE)
		OFF	Has Sleep	
RUN	Running state	1Hz	Be Working…	Only DLSx1(LET)
		OFF	Has Sleep	
SIG	LTE signal	ON	Searching the network	Only DLSx1(LET)
		0.6Hz	Registered network	
		2.5Hz	The data link is established	
		OFF	LTE not working	
LDT	LoRA datas	1Hz	Not sleep	
		Flash	Receiving data	
		ON	Sending data	
		Short	Detecting wake-up signal	

INTERFACE DEFINITION

Pin	Symbol	Name	Descr
1	VIN	External power input +	DC5.5V~24V
2	GND	Power input -	
3	GND		
4	TXD/A	RS232/TTL Send Pin /RS485(D+)	Default TTL
5	RXD/B	RS232/TTL Receiv pin /RS485(D-)	

1. START To USE

1.1 Install Batteries/Connect External Power

The DLSxx operates using a built-in battery and/or an external power supply.When connected to an external power supply, the DLSxx is powered entirely by the external power supply, which also charges the built-in battery.

The methods and steps for installing the built-in battery are as follows: Open the cover of the DLSxx device, install the battery in the battery holder, and pay attention to the positive and negative of the battery.

1.2 Digital Interface

DLSx0 has four digital interfaces (UART-A, UARt-B, LORa-A, lora-B). Any digital interface can complete parameter reading and modification, data forwarding.Each digital interface has a 200-byte receive cache.

Note that the communication is possible only when the interface parameters are the same.UART interface parameters include Baud Rate, Data bit, Parity bit, and Stop bit. LoRA interface

parameters include radio frequency (channel), Spreading Factor, Coding Rate, and Bandwidth.
DLS communication interface default parameters

Interface	Parameter Name	Default	Unit
UART	Baud Rate	115200	bps
	Parity bit	N	
	Data bit	8	bits
	Stop bit	1	bits
LoRA	Channel Num	7 (434/868MHz)	
	Spreading Factor	8	
	Coding Rate	2	
	Bandwidth	7	

The DLS10 has one LoRA and two UART interfaces.The time division multiplexing (TDM) technology to expand the one LoRA to channel A and channel B (LORa-A and LORa-B). Lora-A and Lora-B share the LA_SF, LA_CR, LA_BW, and LA_POW parameters.See "Registers (Parameters) summary" for details.

The DLS20 has two LoRA and two UART interfaces. Lora-A and Lora-B each have itself LxSF, Lx_CR, Lx_BW, and Lx_POW parameters.See "Registers (Parameters) summary" for details.

1.3 Device Information

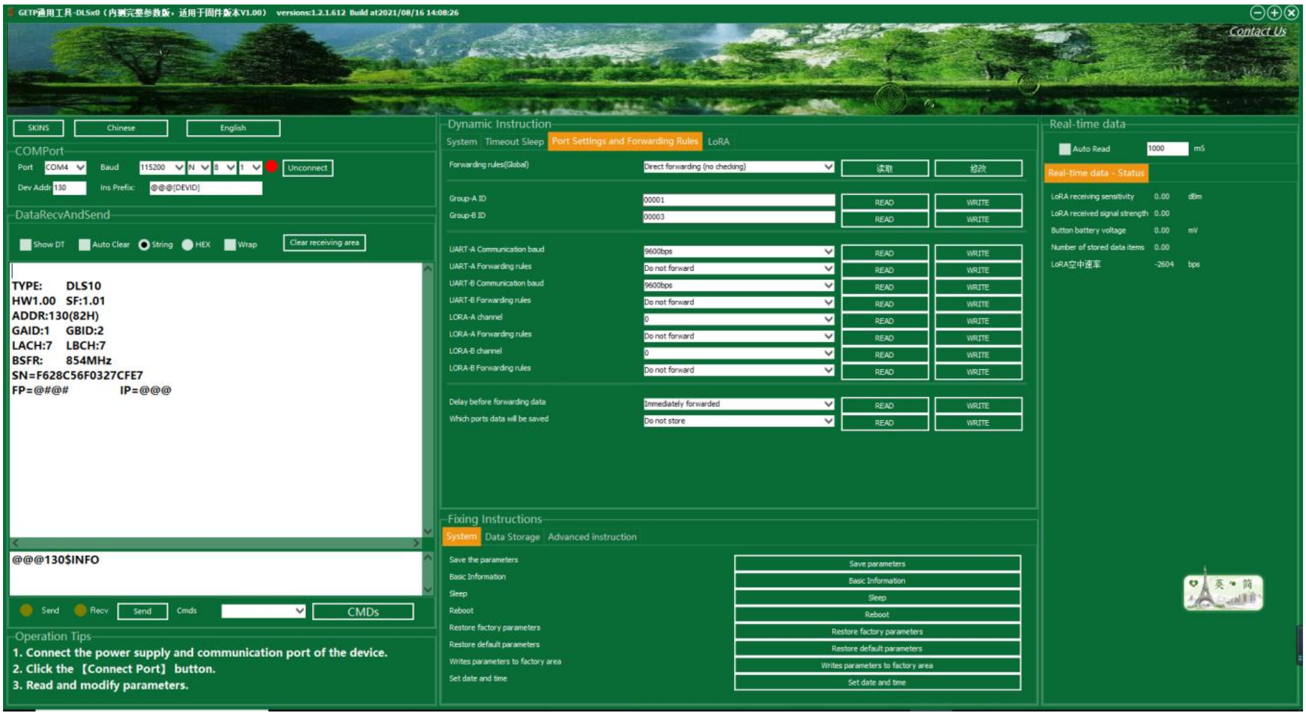
When the device is powered on and started, DLSx0 automatically outputs the basic device information through UART-A. When the device is running, Used "$INFO" to get the device information. The basic information is as follows:

TYPE: DLS10	Device Model
HW1.00 SF:1.00	Hardware version, Firmware version
ADDR:129(81H)	Device address
GAID:1 GBID:2	Group ID A、Group ID B
LACH:7 LBCH:7	Channel number of LoRA-A、LoRA-B
BSFR: 854MHz	RF base frequency
SN=F628C56F0327CFE7	Device serial number (UNIQUE IDENTIFIER)
FP=@#@# IP=@@@	Data forwarding prefix symbol, string instruction prefix symbol

1.4 Use $SETPTool read and write parameters

$SETPTool is a general equipment testing, parameter reading and writing tools, suitable for most of our company's equipment.If you want to write your own test tool, refer to the section "2. Communication Protocols".

The home screen of "$SETPTool for DLSx0" is as shown in the following figure.



Parameters can be read and modified as prompted in the lower left corner of the main interface, and real-time data of the device can be read automatically.For more detailed instructions on how to use $SETPTool, see the "General parameter configuration Tool SETPTool Instructions.pdf" file.

1.5 DLSx0 Work Mode

1.5.1 Work Mode

DLSx0 has two working modes: real-time receive and timeout sleep. When register WKMOD is 0, it works in real-time receive mode, and when register WKMOD is 1, it works in timeout sleep mode.

- **Real-time Receive Work Mode:** The LoRA of the DLS device is in the receiving state.In this mode, only LORa-A is valid for the DLS1x device, while both LoRA channels of the DLS2x are valid.
- **Timeout Sleep Work Mode:** In this mode, when no operation exceeds a predetermined time, the device enters the idle state to save power, and further enters the stop state if there is no data interaction for a long time.The stop state has the lowest current consumption.

1.5.2 State Machine

There are three state machines (standby, idle, and down) for devices that work in timeout sleep mode.

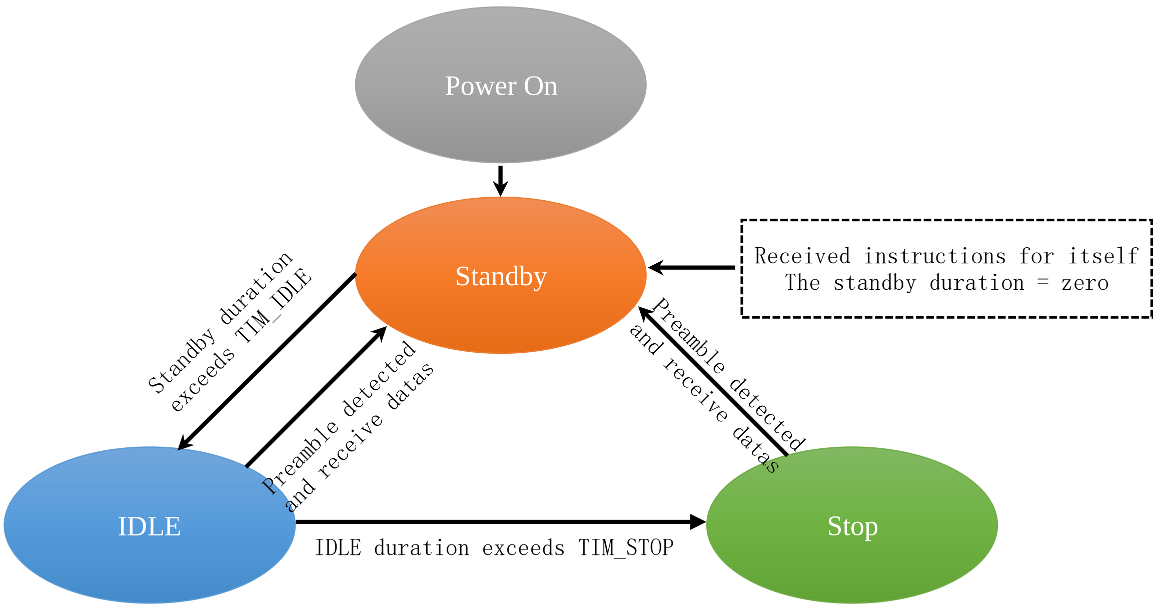
- **Standby State:** In this state, DLSx0 continuously detects whether lora-A and Lora-B have received valid Preamble① signals at an interval of about 10mS. When receiving signals, DLSx0 switches to the receive mode until the data is received.
- **IDLE State:** If the running duration of a device in standby state exceeds TIM_IDLE, the device automatically enters IDLE state. The IDLE state has a lower current consumption. In this state, the UART works normally, and LoRA periodic detection. The LoRA preamble

is detected every time specified by TIM_WUT. When receiving UART data or LoRA preamble code, exit the IDLE state and enter the standby state.

- **Stop State:** When the running duration of the idle device exceeds the specified period of TIM_STOP, the device automatically stops. The Stop state has the lowest current consumption. In this state, the UART power saving, and LoRA periodic detection. The LoRA preamble is detected every time specified by TIM_WUT. When receiving UART data or LoRA preamble code, exit the Stop state and enter the standby state. It is important to note that the first packet of UART data received by the device in the Stop state will be incomplete and the incomplete packet will be discarded without any processing.

When the device in idle or stop state is woken up, the device processes the received data (forwarding, storing, executing instructions, etc.), and immediately enters the idle state again after the processing. If the received data is the instruction for the device itself, the device executes the instruction, automatically switches to the standby state, and sets the total duration of the standby state to 0 (that is, the device enters the idle state again after waiting for TIM_IDLE).

① Preamble: A sequence of synchronization signals actively sent by the sender before LoRA communication, followed by the actual data to be sent.



DLS10/20 state machines

1.5.3 Preamble

Preamble are a necessary part of LoRA's wireless technology. A complete communication process is initiated by the sender, who first sends a number of preamble. When the preamble is detected, the receiver synchronizes its clock and prepares to start receiving data. The LoRA protocol stipulates that the number of preamble should not be less than 6.

A DLSx0 device in real-time receiving mode can receive any length preamble. Therefore, there is no requirement for the number of preamble of the sender (at least 6). When the device is working in timeout sleep mode, LoRA periodically detection for the preamble. In the standby state, the detection is interval of about 10mS, In the idle or stop state will only a short detection every time specified by TIM_WUT. Therefore, the sender needs to send enough preambles (sufficient duration) before sending data. In the DLSx0 device, register TIM_WUT is used to set the detection interval, and TIM_SPB is used to set the duration of sending the preambles. For DLS devices in the IDLE or stop state able to detect the preamble, the TIM_SPB of the sender must not be shorter than the TIM_WUT of the receiver.

The default value of TIM_WUT is 1000mS, and that of TIM_SPB is 2000mS.

1.5.4 Time Parameters

The DLSx0 device has multiple time registers (parameters), described as follows:

TIM_IDLE: For details, see “State Machine”.

TIM_STOP: For details, see “State Machine”.

TIM_WUT: For details, see “Preamble”.

TIM_SPB: For details, see “Preamble”.

TIM_WRD: Timeout for LoRA to receive data. It starts when the first preamble is detected. 0 indicates that the value is automatically set to (TIM_SPB+256 bytes transfer duration) x 2.

TIM_FDL: Delay forwarding after receiving data.

1.5.5 Work mode performance comparison

Work Mode	real-time receiving	Timeout Sleep (Never IDLE)	超时休眠 (IDLE)	超时休眠 (Stop)
conditions parameter	WKMOD=0	WKMOD=1	WKMOD=1	WKMOD=1
	TIM_IDLE=xx	TIM_IDLE=0	TIM_IDLE>0	TIM_IDLE>0
	TIM_STOP=xx	TIM_STOP=xx	TIM_STOP=0	TIM_STOP>0
current	25mA	25mA	3.3mA	8uA
LoRA Number of CHs	1	2	2	2
Detection interval	continuous	20mS	TIME_WUD	TIME_WUD
advantages	Best timeliness	Dul Channels good timeliness	Energy saving	Very energy saving
disadvantages	Power consumption big only 1 LoRA	Power consumption big	poor timeliness	Poor timeliness The first packet received UART was invalid
Requirement for the sender’s preamble	None	>20mS	>TIME_WUD	>TIME_WUD
The xx parameter indicates not care. To improve the timeliness of data transmission, set the TIM_SPB parameter based on the receiver’s requirements on the preamble length				

1.6 LoRA channel and center frequency

DLS uses the channel number to set the RF center frequency.

$$\text{Center frequency MHz} = \text{Base Fre } MHz + (ChNum * 2MHz)$$

DLS LoRA base frequency has been set to 420 or 854MHz before delivery. You can use $STRF=XXX to reset the base frequency.

For example: $STRF=420

Note: The base frequency of the LoRA should not be modified (except to match LoRA from another manufacturer or for very specific purposes).

1.7 Read data from internal storage

Use the dedicated $GTDA=XXXX to get the data with the specified number that has been stored. For details about the protocol of the returned packet, see section “2. Communication Protocol”.

2. COMMUNICATION PROTOCOL

2.1 Registers (parameters) summary

2.1.1 Registers (read/write)

Register summary Table (read/write)

Reg Addr	Symbol	Name	Value Range	Default	Unit	Descr
0	DEV_ID	Device Address	1~255	129		
1	GROUP_A	Group ID A	1~255	1		
2	GROUP_B	Group ID B	1~255	2		
3	WKMOD	Work Mode	0: real-time receiv 1: Timeout sleep	1		
5	TIM_IDLE	Timeout of IDLE	0: Never 1~65535	10	seconds	valid only in timeout sleep mode
6	TIM_STOP	Timeout of Stop	0: Never 1~65535	10	seconds	
7	BBAT_CHG	charge the button battery	0: Never 1: charge on low vol	1		
8	BBAT_LOW	Button battery low voltage	0~65535	2400	mV	
9	FW_RULE	Forwarding Rules (global)①	0~7	7		
10	UA_BAUD	UART-A Baud rate②	12~4608	1152	100bps	
12	UA_FWR	UART-A Forwarding Rule③		0x10		
15	UB_BAUD	UART-B Baud rate②	12~4608			
17	UB_FWR	UART-B Forwarding Rule③		0x40		
21	LORA_SF	LoRA Spreading Factor	6~12	8		
22	LORA_CR	LoRA	1~4	2		

TEL: 0316-3093523 010-61591202 Email: INFO@GEO-INS.COM

		Coding rate				
23	LORA_BW	LoRA Bandwidth	0~9	7		
25	LORA_POW	LoRA Send Power	0~15	10		
24	LA_CH	LoRA-A Channel Num	0~15	7		
26	LA_FWR	LoRA-A Forwarding Rule③		0x01		
34	LB_CH	LoRA-B Channel Num	0~15	7		
36	LB_FWR	LoRA-B Forwarding Rule③		0x04		
40	TIM_WUD	Detection interval	0: Never 50~65535	1000	mS	valid only in timeout sleep mode
41	TIM_WRD	Receive timeout	0~65535	10000	mS	
42	TIM_SPB	Sending duration of Preambles	0~65535	2000	mS	
43	TIM_FDL	Data forwarding delay	0~65535	100	mS	
44	DAT_SAVE	Whether to save data ④		0x0000		
46	DBG_MSG	Output debug message	0: No output 1: Output	0		

(1) Forwarding rule (global) register

bits	Name	Descr
bit15:3	Reserve	
bit2	Whether to check the validity of the destination address before forwarding packets	Invalid packets will not be forwarded. See "Data Forwarding Protocol" for details.
bit1	Whether to check the validity of the forward prefix checksum before forwarding packets	
bit0	Whether to check the validity of the group ID before forwarding packets	

(2) UARt-X communication rate register

bits	Name	Value	Unit
bit15:14	Parity bit	0: None	
		1: Odd	
		2: Even	
bit13:0	Baud Rate	12~4608	100bps

(3) Port forwarding rule register (target port setting)

Each digital interface has an xxxx_FWR(Forward Rule) register that defines which digital interface (or interfaces) to Forward data packets to after receiving them. Each of the two bits of the register goes from low to high to represent a target port. Bit0/1 is UART-A, Bit2/3 is UART-B, Bit4/5 is LoRA-A, and bit6/7 is LoRA-B.In each two bits, the low bit indicates whether the data is forwarded to the port, and the high bit indicates whether the data is output with the Forward Prefix (FP).

Data forwarding register xxxx_FWR

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Forward to UART-A		Forward to LoRA-A		Forward to UART-B		Forward to UART-A	
With FP	Output	With FP	Output	With FP	Output	With FP	Output

Example:

Set UA_FWR is 00010000B (0x10), When UART-A receives the data, forwards to LoRA-A, without FP.

Set LA_FWR is 00000001B (0x01), When LoRA-A receives the data, forwards to UART-A, without FP.

(4) Data SAVE register DAT SAVE

bits	Name	Descr
bit15:4	Reserve	
bit3	Whether to store data received by the LoRA-B	If the data is instructions to oneself, not stored
bit2	Whether to store data received by the LoRA-A	
bit1	Whether to store data received by the UART-B	
bit0	Whether to store data received by the UART-A	

2.1.2 Registers (only read)

Register Summary Table (read only)

Reg addr	Symbol	Name	Value Range	Unit
70	RTC_YM	RTC-Year and month	BCD code	
71	RTC_DH	RTC-day and hour		
72	RTC_MS	RTC-minute and second		
74	RF_BAUD	Air rate ①	12~37500	bps
75	SENS	LoRA receive sensitivity		
76	RSSI	LoRA signal strength		
79	VCC_MV	Working voltage		mV
80	BBAT_MV	Button battery voltage		mV
91	SAVED_NUM	Number of stored data items		

(1) Air rate register RF_BAUD (read only)

The value of this register is the LoRA transmission rate calculated based on the Spreading Factor, coding rate, and bandwidth.

2.2 Parameter read and write protocol

DLSx0 supports device address-based MODBUS, custom AABB, and string instruction set protocols.

2.2.1 MODBUS Protocols

DLSxx supports MODBUS 03, 04, and 06 instruction codes.

(1) 03 (0x03) /04 (0x04) instruction code: read multiple consecutive registers. The instruction format is as follows

Instruction data frame structure

Dev Addr	Function Code 0x03	Start Addr	Regs Count	CRC
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

The returned data frame structure

Dev Addr	Function Code 0x03	Datas Lenth	Datas	CRC
1 Bytes	1 Bytes	2 Bytes	n Bytes	2 Bytes

Example: read the register whose device address is 129, register start address is 0, and read 10 registers continuously

Host sending instruction (HEX): 81 03 00 00 00 0A DA 0D

DLSxx return: 81 03 14 00 81 00 01 00 02 00 01 00 01 00 0A 00 0A 00 01 09 60 00 07 A1 EF

When reading multiple registers, do not read more than 32 registers at a time. Do not attempt to read a register that does not exist.

(2) 06 (0x06) instruction code: modify the value of a single register, the instruction format is as follows

Instruction data frame structure

Dev Addr	Function Code 0x06	Reg Address	Reg Value	CRC
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

The returned data frame structure

Dev Addr	Function Code 0x06	Reg Address	Reg Value	CRC
1 Bytes	1 Bytes	2 Bytes	2 Bytes	2 Bytes

Example: Example: change the value of register 8 in device address 129 to 2400

Host sending instruction (HEX): 81 06 00 08 09 60 11 B0

DLSxx return: 81 06 00 08 09 60 11 B0

(3) Check code calculation

CRC16 - MODBUS algorithm:

```
unsigned int crc16(unsigned char *dat, unsigned int len)
{
unsigned int crc=0xffff;
unsigned char i;
while(len!=0)
{
crc^=*dat;
for(i=0;i<8;i++)
{
if((crc&0x0001)==0)
crc=crc>>1;
else
{
crc=crc>>1;
crc^=0xa001;
}
}
```

```
len-=1;
dat++;
}
return crc;
}
```

2.2.2 AABB Protocols

Read register

	Frame head	Dev Addr	Reg Address	Checksum
HEX	AA BB	1 Bytes	1 Bytes	1 Bytes

Return

	Frame head	Dev Addr	Reg Address	Reg value	checksum
HEX	AA BB	1 Bytes	1 Bytes	2 Bytes	1 Bytes

For example, read the value of register 10 on the DLS device whose address is 129.

Send (HEX): AA BB 81 0A F0 to the device

Device returns (HEX): AA BB 81 0A 04 80 74, 0x0480 is 1152 in decimal.

Modify register

	Frame head	Dev Addr	Reg Address	Reg Value	checksum
HEX	AA BB	1 Bytes	1 Bytes |0x80	2 Bytes	1 Bytes

Return

	Frame head	Dev Addr	Reg Address	Reg Value	checksum
HEX	AA BB	1 Bytes	1 Bytes	2 Bytes	1 Bytes

For example, change the value of register 10 of the DLS device with address 129 to 1152 (0x0480).

Send (HEX): AA BB 81 8A 04 80 F4

Device returns: AA BB 81 0A 04 80 74

Note: The AABB protocol can only access registers with addresses 0 to 127.

Note: in AABB protocol, 0xFF is a universal address.

2.2.3 Device address-based string protocol

(1) Read register

Fixed prefix	Dev Addr	Instruction	Reg Address	End
@@@	xxx	$GETP=	xxx	\r\n

Fixed a prefix: The value is fixed as "@@@".This prefix can be modified using a special instruction, see "Other instructions."

Dev Addr: It is a three-character device address, for example, “129” or “001”.

Instruction: fixed as “$GETP= “.

Reg Address: The register address represented by a numeric characters.

For example, read the value of register 10 in DLSx0 whose device address is 129

Send to DLS: @@@129$GETP=10

DLS return: $REG[10]=01152

(2) Modify register

Fixed prefix	Dev Addr	Instruction	Reg Address	Reg Values	End
@@@	xxx	$SETP=	xxx	, xxx	\r\n

Fixed a prefix: The value is fixed as "@@@".This prefix can be modified using a special instruction, see "Other instructions".
Dev Addr: It is a three-character device address, for example, “129” or “001”.
Instruction: fixed as “$SETP= “.
Reg Address: The register address represented by a numeric characters.
Reg Values: The register value represented by numeric characters.

For example, change the value of register 10 in DLSx0 of device address 129 to 96.
Send to DLS: @@@129$SETP=10,96
DLS return: OK

(3) Read multiple registers

Fixed prefix	Dev Addr	Instruction	Start addr	Regs Count	End
@@@	xxx	$GETP=	xxx	, xxx	\r\n

For example, read the DLSx0 register whose device address is 129. Start from register 0 and read 10 registers
Send: @@@129$GETP=0,10
DLS returns: $REG[00]=00129,00001,00002,00001,00001,00005,00005,00001,03200,00000

(4) Modify multiple registers

Fixed prefix	Dev Addr	Instruction	Reg Address	Reg Values	End
@@@	xxx	$SETP=	xxx	, xxx, xxx, ……	\r\n

For example, modify five registers starting from register 0 in DLSx0 of device 129
Send: $SETP=0,129,1,2,1,1
DLS returns: OK

2.2.4 Other Instruction

Instruction format (structure)

Fixed prefix	Dev Addr	Instruction string	End
@@@	xxx	xxxx	\r\n

Instruction	Functional Description
$INFO	Read basic device information
$REST	Restart
$SLEP	Enter sleep mode (IDLE)
$RSTP	Restore factory parameters
$STFC	Writes the current parameters to the factory area
$STDF	Restore the default parameters
$GTDA=xxx	Read the stored data with record number xxx
$STNM=0	Clear all stored data

$STDT=xxxx	Set the date and time.Example: $STDT=2015/12/21 18:37:05
$STFP=xxxx	Set the forwarding prefix. xxxx is 4 characters
$GTFP	Read data forwarding prefix
$STIP=xxx	Set the instruction prefix. xxxx is 4 characters
$GTIP	Read instruction prefix
$STRF=xxx	Set the LoRA base frequency.For example: $STRF=420, $STRF=854

2.3 Super Instruction @REST

The super instruction is a string instruction without a device address. Use only when the device fails to communicate properly, the instruction prefix is incorrect, or other unforeseen errors cause the device to fail to work properly. After the command is sent to the DLS device, the device restores the parameters to the default, rewrites the factory parameter area, rewrites the instruction prefix, and rewrites the forwarding prefix. After this instruction is executed, DLS will automatically restart.

2.4 Data forwarding protocol

Data forwarding refers to when a certain digital interface receives the data, which digital interface (or which digital interfaces) sends the data out again, thus realizing the function of the repeater.The DLS has four digital interfaces: UART-A, UARt-B, LoRA-A, and LorA-B.

2.4.1 Forward packet structure

Different DLSX0s use specific packet structures to complete the directional transmission of packets. A complete packet consists of two parts: the data forwarding prefix (FP) and the data to be forwarded.

Data forwarding prefix (FP) is a fixed format prefix that contains data forwarding routing information, including the sender address, receiver address, group ID code, and packet length etc. **Generally, The FP is generated and maintained by the DLS device and users do not need to care about it.**

DLS FP structure

forwarding prefix FP								Datas
FP Head	GroupID	Source Device Address	Target Device Address	Forwarded Counter	Forwarded path	checksum	Datas lenth	XXXXXXXX
@#@#	1 Bytes	1 Bytes	1 Bytes	1 Bytes n	n Bytes	1 Bytes	1 Bytes (m)	m 字节

FP Head: @#@#, identify the key information of the FP.
GroupID: Group ID value when the last data was sent.
Source Device Address: Source of the packet (device address).
Target Device Address: Target of the packet (device address).
Forwarded Counter: The number of times the packet has been forwarded (already n times).
Forwarded path: N bytes. Each byte in turn stores the address of the device that forwards the packet.
Checksum: The sum of all the previous data.
Datas lenth: The actual length of transmitted data.

2.4.2 Data Forwarding Rules

(1) Each device in an area has a unique device address (devices with the same address are not allowed).The valid device address ranges from 0x01 to 0xFE.0xFF is a universal address, and 0x00 is a nonexistent address.
(2) Each device has two UART (UART-A/ UARt-B) and two LoRA (LORa-A/Lora-B) interfaces.Each device has two group ID (GroupA_ID/GroupB_ID, GAID and GBID for short). Each interface can be individually configured to forward data to another interface (output) after receivd.
(3) The packet to be transmitted consists of forwarding prefix (FP) and the data to be forwarded. FP contains the Group ID, Source Device Address, Target Device Address, Data forwarding path etc. The DLS device automatically generates FP for packets without FP. Automatically generates FP rules is:
Group ID= Group ID corresponding to the digital interface.
Source Device Address =0xFF, Target Device Address =0xFF.
(4) After receiving the data packet, DLS will judge whether the Group ID in FP is the same as one of its own two Group IDs (match). If the Group ID matches, DLS will forward the packet to the specified interface (output). The group ID is automatically updated according to the output interface when packets before forwarding.
If the packet is itself instructions, only the instructions are executed (not forwarded).
If the destination address in the packet is 0x00, the packet is not forwarded.
(5) When the target address in the packet is the same as itself address, the target device address is automatically changed to 0x00 during forwarding.

Example: Hexadecimal data packets 40 23 40 23 02 81 85 02 81 82 D3 03 31 32 33
40 23 40 23: FP Head @#@#
02: The Group ID 02 was used when the packet data was last sent.
81: This packet is sent from a device with address 0x81.
85: The target address of this packet is 0x85.
02: This packet has been forwarded 02 times.
81 82: This packet is first forwarded by device 0x81 and the last forwarded by device 0x82.
D3: The sum of all the previous bytes.
03: The actual content of this packet data is 03 bytes.
31 32 33: The actual data content of this package is 0x31 0x32 0x33。

2.4.3 LoRA Data Receive Mapping

When LoRA-A and LoRA-B have the same channel, both channels may receive LoRA packets. In order to maintain the correctness of data forwarding logic, DLS first performs the following processing after receiving LoRA packets.

- Check whether the packet has the correct FP Head symbol (default: "@#@#")
  - Yes: Check whether the group code in FP matches.
    - Yes: Map the data to loRA-X corresponding to the Group ID.
    - No:
  - No: Map the data to LoRA-A.

LoRA receiving data maps means:

Whether LoRA packets are received by LoRA-A or LoRA-B, the received packets are forced to be updated to the receive cache of LoRA-A or LoRA-B according to the above rules, and then processed according to the data forwarding rules.

2.5 Data Storage Protocol

The register DAT_SAVE is used to set whether to store the data to the internal Flash chip after the digital interface receives the data.Bit0 stands for UART-A, bit1 stands for UART-B, bit2 stands for LoRA-A, and bit3 stands for LoRA-B.

For example, DAT_SAVE=0x0001 Indicates that data is stored in the Flash chip after UART-A receives the data. DAT_SAVE=0x000F indicates that data received by any digital interface is stored.

You can use the "$GTDA=data number,data protocol code" command to read the stored data packet. When the data protocol code is 0, it indicates that the data is output without basic information; when the data protocol code is 1, it indicates that the data is output with basic information.The output data format of DLS is as follows:

Record identification	Basic information	Datas
DATxxxx=	XXXXXXXX	XXXXXXXX

Basic information									
packets size	Date Time	Port Code	Group ID	Source address	Target address	Forwarded Counter	Forwarded path	checksum	Datas lenth
1 Bytes	6 Bytes	1 Bytes	1 Bytes	1 Bytes	1 Bytes	1 Bytes n	n Bytes	1 Bytes	1 Bytes

Record identification: Fixed to “DAT+4 digit characters”, The digit is the record number.
Packets size: The total number of bytes including this byte.
Date Time: : BCD Indicates the year, month, day, hour, minute, second.
Port Code: Which digital interface receives and stores the data.0: UART-A, 1: UART-B, 2: loRA-A, and 3: LoRA-B.
Group ID, Source address, Target address, Forwarded Counter, Forwarded path, checksum has the same meaning as FP, see section "Forward Packet Structure" for details.

For example,
Send to DLS: @@@129$GTDA=79,1
DLS return (HEX) : 44 41 54 30 30 37 39 3D 14 16 01 12 01 45 17 00 01 FF FF 00 C5 06 31 32 33 34 35 36
44 41 54 30 30 37 39 3D: DAT0079=
14: The total number of bytes including this byte, 0x14=20 Bytes.
16 01 12 01 45 17: The storage time of this data is 2016-01-12 01:45:17
00: This data is received and stored by UART-A
01 FF FF 00 C5 06: The FP when this data is received. 0x06 indicates that the actual data

content is 6 bytes. For details, see "Forward Packet Structure".
31 32 33 34 35 36: The actual data content.

For example,
Send to DLS: @@@129$GTDA=79
DLS return（HEX）: 31 32 33 34 35 36

Note: When reading the data recorded as 0, DLS will return the last stored data, for example, send "@@@129$GTDA=0,1" to the device, and the device will return "DAT0123=......"., which indicates that the last record number is 123.

3. DLS APPLICATION EXAMPLE

The following Application Example (AN) are based on the DLS default parameters.

3.1 UART (RS232/RS485)、LoRA Conversion

This application example realizes the wireless transparent transmission between two UART devices. Device I and device II are UART interfaces.

DLS (1#) Parameter setting: Set UART-A communication parameters consistent with device I.
DLS (2#) parameter setting: Set UART-B communication parameters consistent with device II.
Note: Set whether to store data as required.

3.2 LoRA Repeater - Using Group ID

This application example realizes the long distance wireless transmission of two UART devices.

Data transmission example:

★Device I sent by UART: "123" (0x31 0x32 0x33)

DLS (1#) received UART-A data "123" and added FP to it. Because the parameter UA_FWR=0x0030, the data received by UART-A is forwarded to LoRA-A with FP. Therefore, DLS (1#) loRA-A output data is: (HEX)

40 23 40 23 01 FF FF 01 81 47 03 31 32 33

DLS(2#) LoRA-A received the data packet, and output by LoRA-B, the sent content is:

40 23 40 23 02 FF FF 02 81 81 CA 03 31 32 33

……

★DLS(4#) LoRA-A received packet 40 23 40 23 03 FF FF 04 81 81 81 81 CF 03 31 32 33, Because LA_FWR=0x0001, the data is forwarded to UART-A without FP. The contents sent by RART-A is 31 32 33, That is, **the UART of device II receives "123".**

For details about the FP, see section "Data Forwarding Protocol".

Similarly, the data sent by the UART of device II can also be forwarded by DLS and finally arrive at device I. The data transmission process is basically the same as described above.

In this example, the Group ID matching and forwarding rule of the DLS device is utilized. When any DLS device sends data through LoRA-x, other devices will receive the data, but only the device matching the Group ID will forward the data packet, thus realizing the directional serial transmission of the data packet and extending the communication distance of LoRA.

The disadvantage of this application example is that when any device sends LoRA, it wakes up all DLS devices at the same time, resulting in a certain amount of power loss.

3.3 LoRA Repeater - Using Different Channel

This application example realizes the long distance wireless transmission of two UART devices.

The data forwarding process of this scheme is exactly the same as that of "3.2 LoRA Repeater - Using Group ID".

The advantages of this scheme are as follows: when a DLS device sends LoRA data packet, only the DLS device with the same channel will be awakened, which realizes data transfer and reduces the power consumption of the whole relay network.

3.4 Modify the parameters of a specified device on the network

This application example changes the value of register 10 of the device whose address is 131 to 1152.

In application example 3.2 or 3.3, set the addresses of devices 1# to 4# to 129, 130, 131, 132 respectively.

(1) Device I is the computer, the computer sends the string instruction "@@@131$SETP=10,1152" to DLS (1#) through the UART interface.

(2) Data is forwarded. DLS(1#)LoRA-A->DLS(2#)LoRA-A->DLS(2#)LoRA-B->DLS(#3)LoRA-A.

(3) DLS (3#) after receiving the instruction to execute, and return "OK".

Because the data content is a DLS (3#) instruction, no further forwarding is required (see "Data Forwarding Protocol").

3.5 Match with LoRA devices from other manufacturers

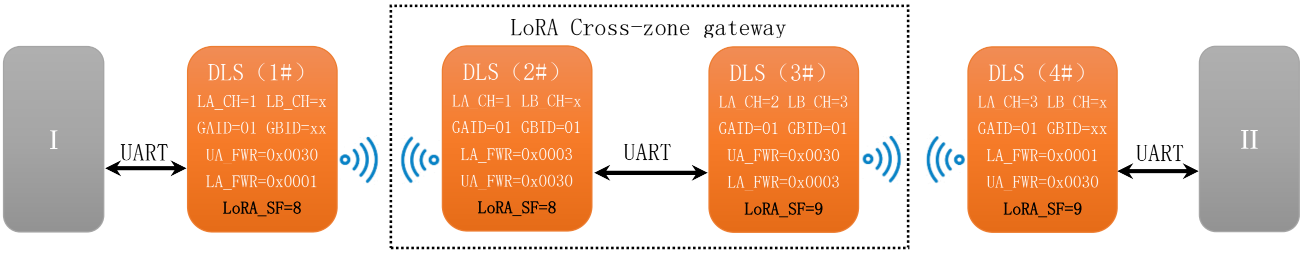
A necessary condition for data transmission between different LoRA devices is complete consistency of communication parameters, including Spreading Factor, Coding Rate, Bandwidth, and center frequency.First get the above four parameter values, and then modify the corresponding register of DLS.

For the center frequency Settings in DLS, see "LoRA Channel and Center Frequency".

If the preamble is less than 50mS when other LoRA devices send data, the DLS must work in real-time receive mode.

3.6 LoRA Cross-zone gateway

DLS10 must use the same spread spectrum factor (SF), coding rate (CR), and channel bandwidth (BW) for data forwarding.A data forwarding network consisting of multiple devices using the same 3 parameters is called a "zone". This application example implements cross-zone data forwarding.

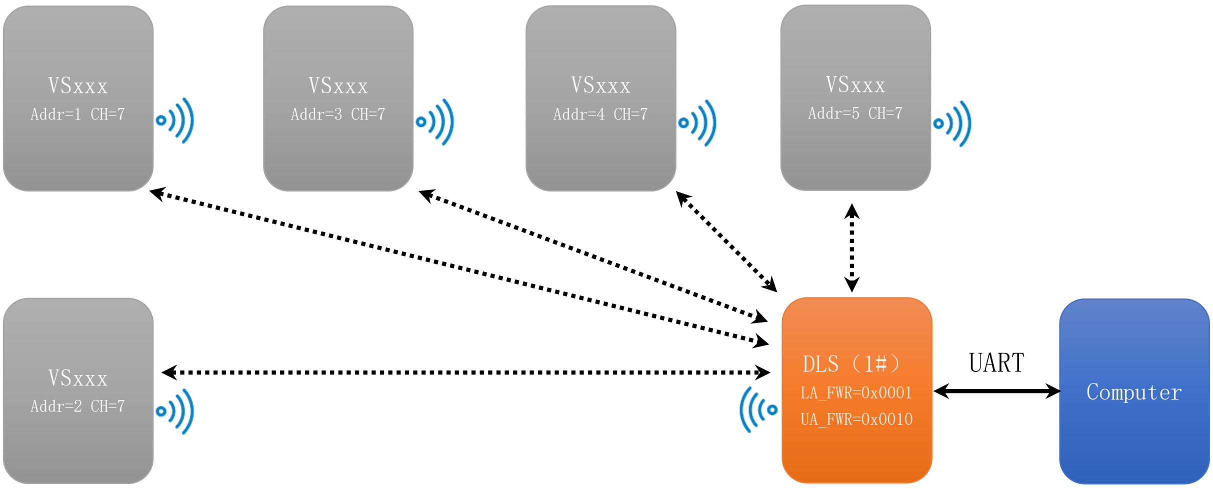


3.7 Use DLS1x to match the LoRA of the VSxxx device

VSxxx is an instrument with LoRA data transmission, and its LoRA default parameters are exactly the same as those of DLS.

The radio parameter register of VSxxx is Spreading Factor (283), Coding Rate (284), bandwidth (285) and Channel Number (286) respectively. It must be ensured that these four parameters are exactly the same as the corresponding parameter value of DLS. In addition, VSxxx's register 289 determines the lead length for LoRA to send data. The default value is 5 seconds (this value should be as small as possible, but not less than DLS TIME_WUD).

(1) Use DLS10 as LoRA receiver for VSxxx



(2) Extend LoRA communication distance of VSxxx

This AN uses several DLS10s to extend the LoRA communication distance of VSxxx equipment.

(3) Data collection and remote transmission of multiple VSxxx devices

This AN builds an LoRA network for one-way data transmission, bringing together data from all VSxxx devices to DLS11, and ultimately forwarding the data to the local computer, other LoRA devices (networks), and remote servers based on an InterNet network.