Quadrep Electronics(T)

QRZ-Stack AP Specification

V2.04

ver03

QuadRep Electronics [T] Ltd.
16F-1, No. 75, Hsin Tai Wu Rd, Sec.1, His-Chih, Taipei, Taiwan
TEL: +886-2-26989933
FAX: +886-2-26989911
http:// www.quadrep.com.tw
http:// www.quadrep.com.cn

Modified record

Version	Content conspectus	Modified person	date
V0.6	Initialize text	Liwei Chour	2006/09/13
V0.7	1.Remove I^2C interface Sensor 2.Modified command	Liwei Chour	2006/09/13
V0.71	1.RAWDATA add SrcDev_ID column	Liwei Chour	2006/09/21
V0.90	1.Command version modify to 0x01 2.Sinplify parameter content 3.Command add length column 4.Add Device quest command	Liwei Chour	2007/2/1
V0.91	1.Add the command for getting the information of child module in the coordiantor 2.UART in charge of Sensor Sensor information	Liwei Chour	2007/4/12
V1.00	1.Formal Release	Liwei Chour	2007/6/26
V2.0	1,Change paramters from the network 2.Modifies the command 3.Remove LEACH mode 4.Add power saving	Liwei Chour	2008/3/13
V2.01	1.Add encryption 2.Add Raw data sendout command 3.Payload of RawData magnify to 60bytes	Liwei Chour	2008/5/23
V2.03 ver1	1,Fix Bug 2.Sleep timer unit changes to ms	Liwei Chour	2008/7/14
V2.03 ver2	1.Adding sample of parameters 2.Adding default parameters 3.Adding use case	Liwei Chour	2008/8/7
V2.03 ver3	1.Fix the bug of Command 0x13	Liwei Chour	2008/9/15
V2.03 ver4	1.Fix Bug 2.Add English version	Liwei Chour	2008/10/3
V2.04 ver1	1. Changing the PING 2. Adding loaction detecting 3. Adding No-confirm Coordinator	Liwei Chour	2008/11/11
V2.04 ver2	1. Adding Fixed parent only 2.The time which to wake up the outside MCU can be changed	Liwei Chour	2008/12/29

1.Bascal setting

Goal fo system

Using RF IC of 802.15.4 to build a network which can send and receive information.

The limitation of system

Changing of program up to parameters, through the command to modify parameter values, not only re-complier program, but also buy ICE.

Allow detailed SDK manual, customer use command to modify program by them.

System allows point by point, tree and mesh network topology.

Noun explain

64 bits Address: The unique address of RF module. We can identify the module using this address. If we repeat address, it will make unexpected mistake. This number is a part of Zigbee network.

16 bits Address: The modules send and receive data using this addess. They aren't same value in a network. This number is a part of Zigbee network.

PANID: The identifiable value make by Coordinator. The modules which have same PANID are in the same Coordinator. This number is a part of Zigbee network.

Channel: The frequency which the modules use to communicae to each other. It should be same between modules. The value of frequency is between 0x00 to 0x0F. This number is a part of Zigbee nertwork.

Outside equipment number: When the module connects with the outside equipment, the outside equipment provides a user-define number, the length of number is 64 bits. We can assign it as goal of transform when we send some information. Thus and so, user have more freedom in sending information. This number is not any repetition.

Coordinator: A center of network, settler of parameter and center of information. There is one Coordinator in a network. Importantly, if Coordinator is wrong, the network stops.

Device: The outside handware can send the information to Coordinator or the other Device.

Fix module: The child classification of the device. The position of this item is fixed. It can receive and re-send the data which come from the other Device.

It can be the Router.

Move module: The child classification of the device. The position of this item can be any where. It can't receive and re-send the data which come from the other Device. It can be the Terminal.

2. Module direction

Device

Goal

It provide a low price, multi-function and low power consumption module. Customers connect reach-me-down sensor or IO module, and we can reach to monitor and control.

Function

It provides 2 IO, 1 digital input, 1 digital output.

It provides 21port of 10bits-AD Converter. Voltage range is from 0 to the working voltage (5.0 V or 3.3V).

It provides the difference baud rate.

It provides the encryption and de-encryption function.

It provides the sleep mode to reduce power usage.

It provides unique 64 bits address.

It provides 1 LED to show the condition and detect the mistake.

Setting paramters

Using the UART or the RF to send the parameters. The parameters arr saved in the FLASH of the Device.

Adopt the UART way to set. First, connecting the module and the RS232 baby board then connects with the computer. After that, turn on the power of the module, and parameters are sended to the module by RS232 to change the parameters.

When adopt RF to send the parameters, we set Coordiantor on the RS232 baby board. Then we connect it with the computer. Turning on the power of Coordiantor and Devices, we send the parameter command to the Coordiantor. Coordiantor send commands to Device using Network automatically, and the Device modifies parameters.

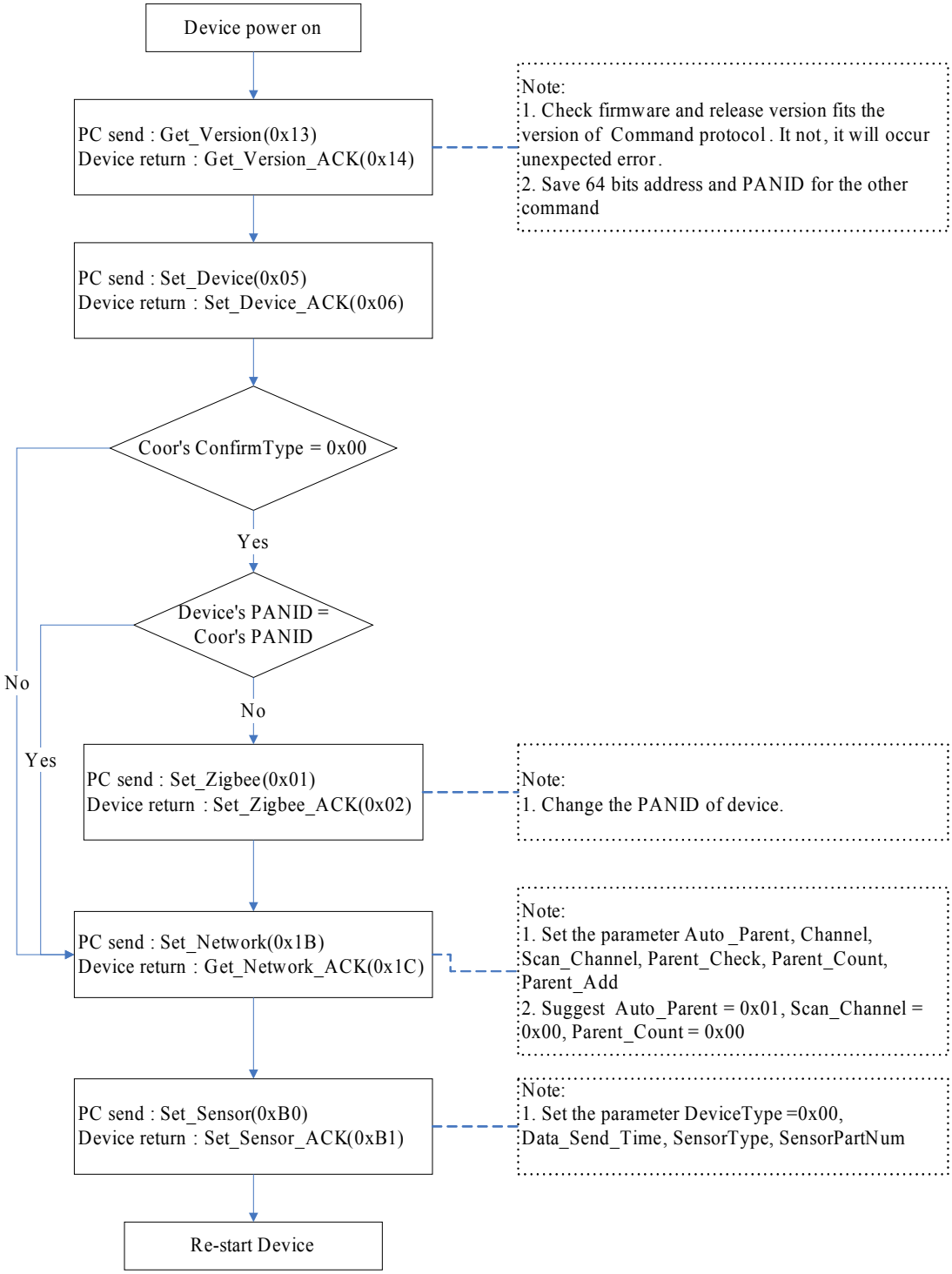
Format of paramters

When using the UART to send the parameters, users should follow the special format, call QR format. Please refer the chapter 4 to get the detial information.

Process of setting parameters

The parematers which the Device need are Device type (command 0x05), the PANID and the channel of the device (command 0x01), building and repairing netowork (command 0x1B), the type of sensor (command 0xB0).

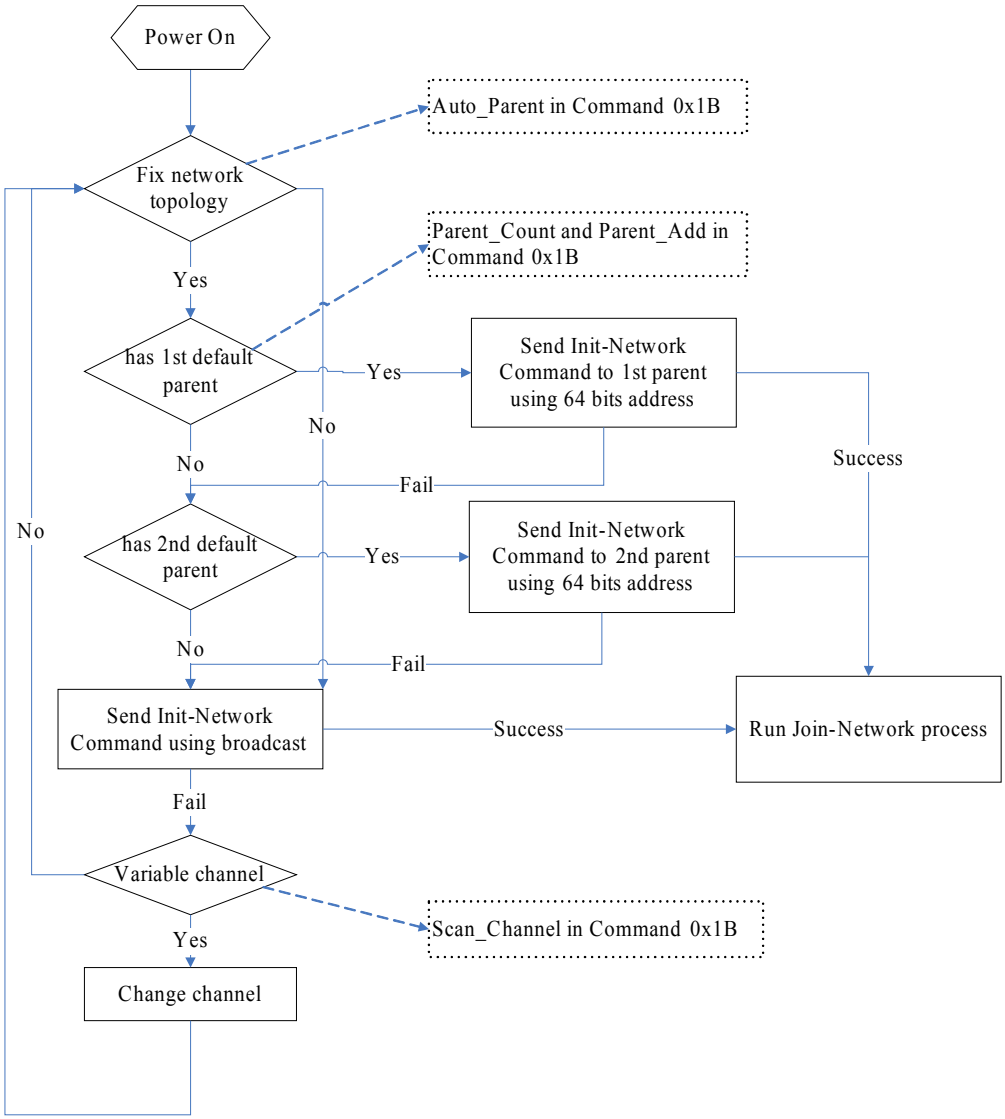
Device Parameters Setting Process



Process of initiaize network

When proceed the work of initialize network, Device use fixed 64 bits address or broadcast to send Device's 64 bits address out, and find a parent to build parameters of the network. If it can't find out a parent in a channel, Device hunts for it over again and it changes channel by itself.

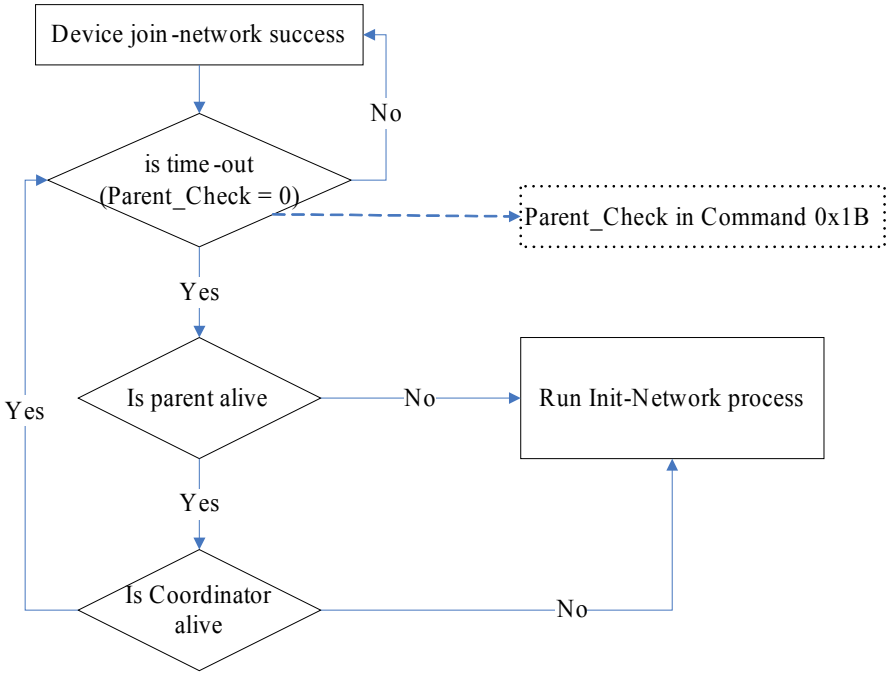
Initialize Process



Process of network servival detecting

Device will connect with Parent periodically to confirm the survival of network. If it detects unsuccessfully, and it procees initialize network again to find a new parent.

Check alive Process



Sending sensor data

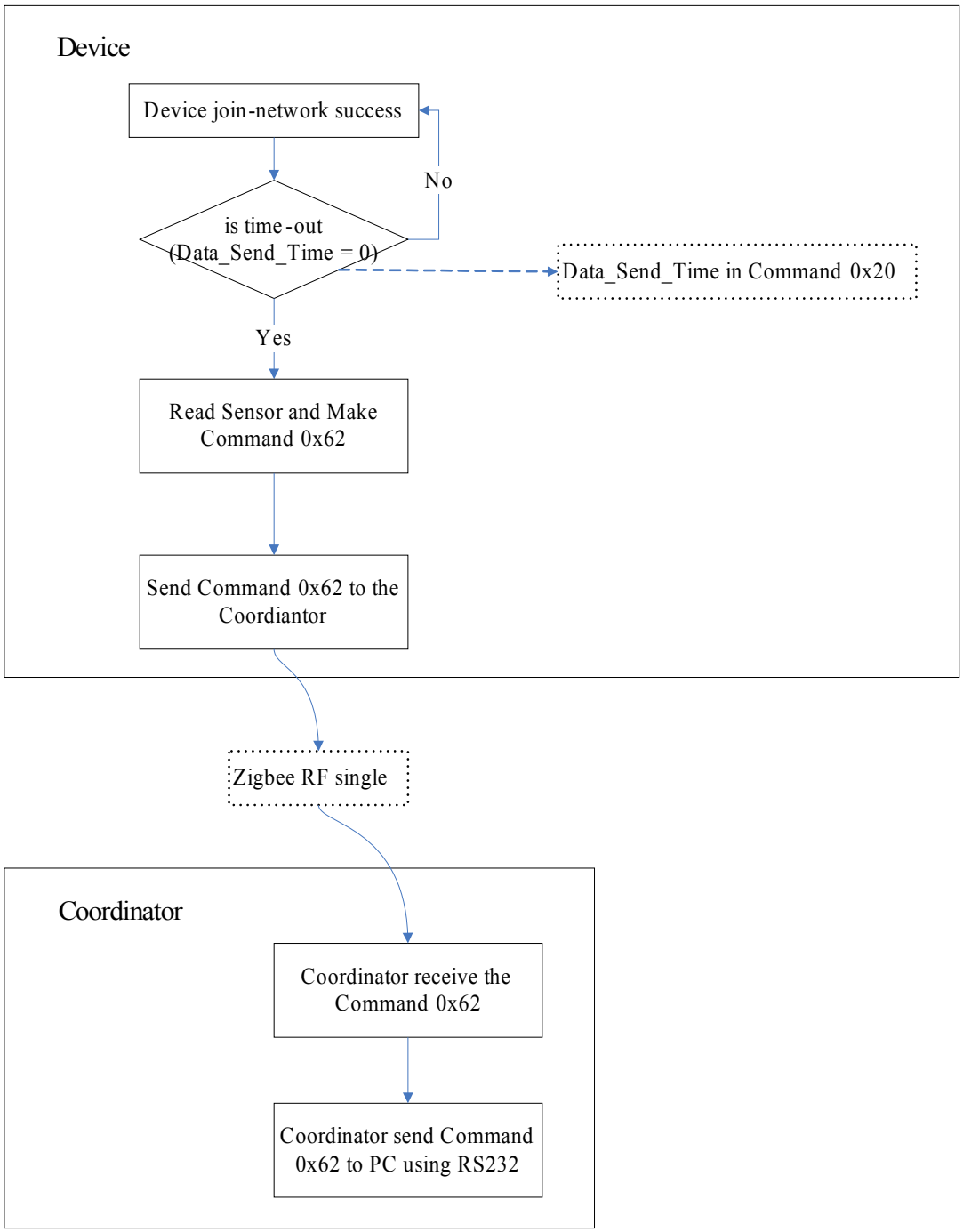
Device sends sensor information by command 0x62 and recive sensor acion by command 0x63. Device sends out command 0x62 actively to Coordiantor. Then Coordiantor send to outside MCU through UART, after it receives. When outside MCU receives 0x62, it puts worked consequence in command 0x63. It sends to Coordinator by UART, and it transships for Device to deal.

Coordiantor can use command 0x64 to ask Device report the status of GPIO.

Transference of GPIO adopts one IO is a value. 1 is High potential, and 0 is Low potential.

Anagle communication returns AD value of sample between 0 and 1024.

Read Sensor Process



Sensor pins assignment

Definition as below, pins of digital input, digital output and AD transform:

Pins name	location	style
digital input 1	GPIO6	Input Only (High-impedance)
AD transform 1	GPIO7	Open-Drain Output
digital output 1	GPIO8	Push-Pull output

User define data

There are two way to send user define data, one is QR format, the other is transparent format.

When using QR format, user should package the data into command 0x67, then send command to Device form UART. Device will send command to Coordiatnor firist, then Coordiantor will base the 64 bits address of destination to send the command to the outside MCU by UART or the other Device by RF.

After receving the command 0x67 from RF, Device send the command to outside MCU by UART certainly.

When using transparent format, the user can send data to device directly, then Device send data to Coordiantor. Coordiantor will package the data into command 0x67 and send out using UART.

Defination of sleep IO

The IO port which used in sleep mode are:

Pins name	location	style
Be waked up the module by outside MCU	GPIO1	Quasi-bidirectional
Wake up outside MCU	GPIO2	Open-Drain Output

Coordinator

Goal

It provide a low price, multi-function module. It can set parameters of Device, receive Sensor data from Device, and transform received to outside equipment.

Function

Provide URAT as an interface with outside equipment for send data.

Provide Device function of initialize network.

Provide the encryption and de-encryption function.

Provide the fix baud(115200 bps, 8 data bits, 1 stop bit, no check)

Provide a dispatch way for PAN ID, mudole number, 16 bits address.

Provide a unique 64 bits RF address.

Provide 1 fixed 16 bits address (must be 0x0000).

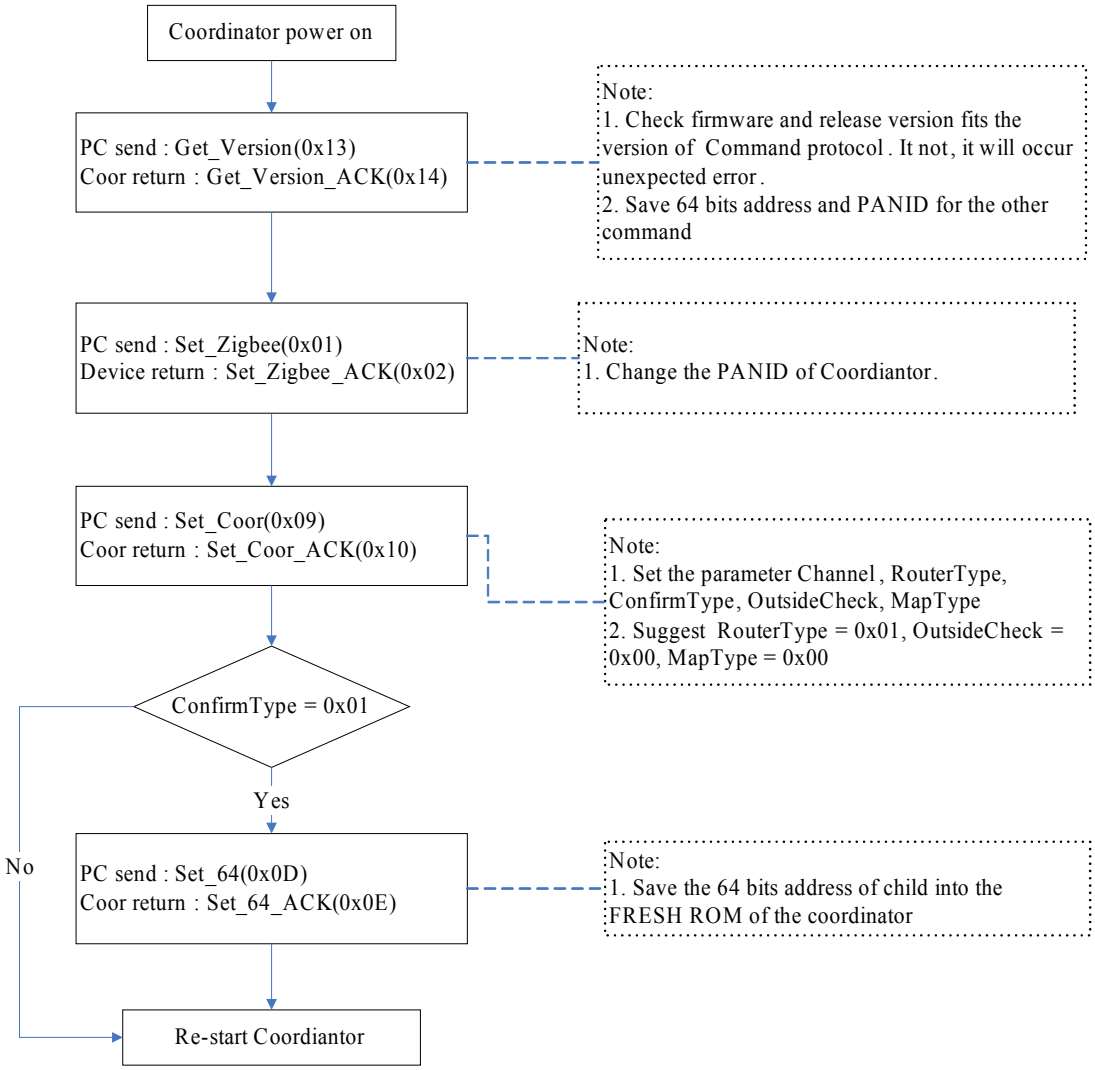
Setting parameters

UART sends parameters to Coordinator. After it receives parameter, it would save parameter in FRESH ROM for next time.

Process of setting parameter

The parematers which the Coordinator need are Coordinator work type (command 0x09), the PANID(command 0x01), store the 64 bits address of child module (command 0x0D).

Coordinator Parameters Setting Process



User define data

User packages the data into command 0x67, then send to Coordiantor by URAT. Coordiantor will base the 64 bits address of destination to send the command 0x67 to the other Device.

3. Direction of system

Working mode

There are tow working modes, one is WSN mode, the other is detecting mode

WSN mode

The WSN mode is the base functin of the modules. In this mode, user can use the auto-building network, auto-repair network and sending data.

Detecting mode

Users can use the detecting mode to decide the location of the modules. In this mode, user sends command 0x88 to get the strength of RF which comes for a special module. So she can decide all the positoins of the modules without install the moduels.

Limitation of connection

One Coordinator connects Deivces most is total 255, and the maximun depth of the network is 15.

Routing method

The method for down-send data uses touting table. The mothod for up-send data is that sending data to the Parent. It will re-send to its parent and up to Coordinator.

Routing table

The Routing table method is that Coordinator save 64 bits address of each module and its completion routing path. When the Coordinator receives a down-send command, it finds the routing path by 64 bits address of the module. It packs routing path and data into a complete down-send command and follows address of routing path to send next one.

The module that receives down-send command checks weather only one routing path or not. If yes, it for itself and decode command. If not, remove first address in routing path. Then, send next one continue.

It saves low byte of 16 bits address in route table, and it need deal 16 bits address of complete.

Movable point and Fixed point

Fixed point is fixed, it will confirm the survival of the network actively. If it can’t find network, it would do initialize netwotk again to find a new network. It can have child and re-send command to the Coordinator. So the Fixed point can be Router. We suggest that setting the module as the Fixed point in the normall case to reduce the difficulties of building network.

Movable point will confirm the survival of the network actively. If it can’t find network, it would do initialize netwotk again to find a new network. It can't have child and can't re-send command to the Coordinator.

Be a attention, the Movable point can't sleep. If you want to use sleep mode, you should set the type of the modules before install the network.

Setting parameter

Setting method

There are two ways to setting the parameters. One uses the UART, the other uses the RF. The Coordinator just uses the RF, and the Device can use RF or URAT.

The UART method also is called as dierct setting. When using the UART method, user should connect module with the setting equipment (like PC) using UART. The setting equipment send the QR command to the module through UART.

The RF method also is called as remote setting. Using the RF setting, user should connect Device with one Coordinator, then send the Device command to Coordinator, Coordinator will send these commends to the Device.

Three setps setting

When changing the parameters using RF method like command 0x17, 0x01, it must be very carefully. If you make some mistake, some mudules will never connect to the network. So the way of chnaging the Zigbee parameters and Fixed parameters is three steps. The first step is that putting the new data into temporary memory, the second step is that saving the data in the temporary memory into FLASH, the thrid step is that re-boot the module.

You can use three steps method directly to update parameters by UART way. If using RF way, we suggest that run step 1 and 2 first then check where the parameters of modules are right or not. After that, run step 3.

Setting Type of Device

Device has two types, one is called RAW Device, the other is Sensor Device. The RAW Device can pass user-defined data. User should pack user-defined data into command 0x67 and send it to the Device using UART. The Sensor Device is not only include the functions of the RAW Device but also reads the status of GPIO and packs into command 0x62.

The Command 0xB0 can set Device as the RAW Device or the Sensor Device.

URAT data sending

QR format

The most useful format is QR format. In this format, you can change the parameters and send data to the other module.

Transparent format

Because the equipment can't be change, user can't make QR format to send data. At that time, user should use transparent format.

When using transparent format, Device will pack the data which comes form outside equipment into command 0x67 and send to Coordinator. When Device receives command 0x67 from Coordinator, Device will unpack the command

0x67 and send the payload to the outside euqipemnt only.

After setting as transparent format, Device can't change parameters through the UART. If user want to change the parameters, she should use the RF way.

UR Baud setting

The command 0x24~-x27 can change the baud of the Device. The command 0x24 is a tow steps functions, User can update FLASH only or using these parameters right now.

Procedure of bulid network

Fixed way

To build fixed way of network topology as below:

1. The network topology parameters are written in Coordinator, Device by the UART.
   a. Using command 0x01~0x04 to get the PANID and the channel of Coordinator.
   b. Check where the PANID and the channel of Device are as same as Coordiantor's or not.
   c. Using command 0x1B~0x1E to set the network parameters of Device. The AutoParant equals 0x00. Putting the default parent's address in the Parent_Add.
2. Setting the way of Device working.
   a. Using command 0x05~0x08 to set the Device parameters. The MoveType equals 0x00, the EXTWAKE equals 0x00.
3. Setting the way of Coordinator working.
   a. Using command 0x09~0x0C to check the identification method. If using 64 bits address identification, user should save the 64 bits address of Devce into Coordinator using the command 0x0d~0x12.
   b. Setting the sleep parameters using command 0x20~0x23. The Wakeup_Time euqals 0x00.
4. Starting Coordinator first, then strating Device orderly by distance. Device runs network initiaize to test exactitude of network.
5. Install Coordinator and Device in position.
6. Start orderly Coordinator and Device over again, and complete build network.

Unfixed way

To build unfixed way of network topology as below:

1. The network topology parameters are written in Coordinator, Device by the UART.
   a. Using command 0x01~0x04 to get the PANID and the channel of Coordinator.
   b. Check where the PANID and the channel of Device are as same as

Coordiantor's or not.

c. Using command 0x1B~0x1E to set the network parameters of Device. The AutoParant equals 0x01.

2. Setting the way of Device working.

a. Using command 0x05~0x08 to set the Device parameters. The MoveType equals 0x00, the EXTWAKE equals 0x00.

3. Setting the way of Coordinator working.

a. Using command 0x09~0x0C to check the identification method. If using 64 bits address identification, user should save the 64 bits address of Devce into Coordinator using the command 0x0d~0x12.

b. Setting the sleep parameters using command 0x20~0x23. The Wakeup_Time euqals 0x00.

4. Install Coordinator and Device in position.

5. Start Device, Device by itself find the Device or Coordinator recently to initiaize network

Sleep

Basic

Because RF IC can't send and receive the single in the sleep mode, so Moved point can sleep. Fixed point can't sleep. If the Moved point wakes up and can't find the parent in time, it will break the network connection.

The process of sleep as blew:

1. Coordinator sends the sleep command to all Devices. The Moved point starts sleep procress after receives the command.

2. In the sleep period, Devices will wake up regularly. After Devices wake up, the first thing is that checking where the parent is alive or not. Then Device reads the working command from his parent. Base on the working command, Device can do the action like leaving sleep, wake up compulsively, reporting the status ...etc. The Sensor Device will send the GPIO to Cooridnator after waking up every time. When wake up time is out, Device will sleep again.

3. If user wants to operator the special module, she can use command 0x8A to wake one module compulsively. When Device be waked up, it will send command 0x8B and keep wake serval seconds.

4. Setting the network in sleep once, Coordinator will send sleep command regularly. So the new device can sleep too.

Suggestion of building network

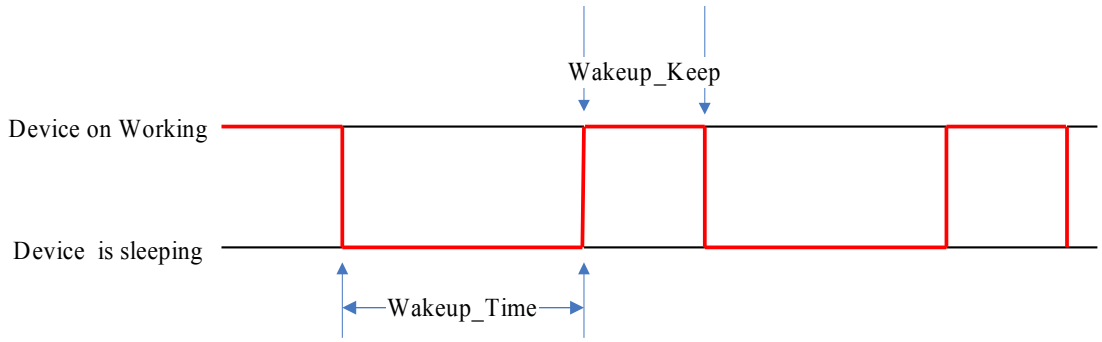
Because the limitation of sleep, we suggest that using the Coordinator and Fixed points to make the main structure. The RF single must cover all space which the Moved point works. Each Fixed point can find 2 parents at least. Then you can put the Moved points which can sleep any where.

When you buliding the main sttucture, the Fixed points can use better

equipment like the extern antenna or PA. It can expand the range of the RF single and reduce the amount of Fixed points. The Moved point can use normal equipment and can expand the life time of the battery.

Coordiantor parameters

In Coodinator side, the sleep commands are 0x20~0x23. The Wakeup_Time sets the time period of sleep, the longer of time period, the less of power usage, but the user should wait more time. The Wakeup_Keep sets the working time after waking up, the longer of working, the less of battery's life. Seting is as 0x02 normally. If you use Sensor Device, we suggest that you set as 0x04.



Device parameters

In Device side, the sleep commands are 0x05~0x08. The MoveType must be 0x01, the Sleep Mode must be 0x00.

Foece sleep and Stop sleep

The System can enter sleep compulsively or leave the sleep using command 0x8C.

Extern interrupt to wake up

Device has an extern interrupt GPIO1 to wake up in sleep mode. The interrupt uses low duty. Using command 0x05~0x08 to decide where extern interrupt works or not and the time period of working time.

Notify outsied MCU

When Device wakes up, it uses GPIO2 to notify outside MCU. The normal status of GPIO2 is high. After wake up, Devie will set GPIO2 to low and keep 5ms. The GPIO2 uses Open Drain to output single.

If Device wakes up base on Coordinator command, it will send command 0x8E to outside MCU.

Encryption

The module offers 7 types of encryption methods. There are AES-CTR, AES-CCM-128, AES-CCM-64, AES-CCM-32, AES-CBC-MAC-128, AES-CBC-MAC-64, AES-CBC-MAC-32. But AES-CBC-MAC-128, AES-CBC-MAC-64 and AES-CBC-MAC-32 don't encrypt the data and jsut add some identification code at the end of data. So wu don't suggest use them.

When using the encryption, the module will add identification code in the end of data. The total length of RF single will be bigger. So if you use encryption, the lenght of payload in command 0x67 is 50 not 60.

Base on the algorithm of deencryption, we don't add encryptio on broadcase way. that means the command 0x66 can't use encryption.

4. Command

Classical command

PC send UART command to modules by RS232, and RS232 parameter are 115200bps, 8 data bit, no check, 1 stop bit (Device) or 9600bps, 8 data bit, no check, 1 stop bit(Device). Beacaue the limitation of MCU, the total length length of command can’t beyond 83bytes.

For distinguish the complete command, every command add fixed Head and Tail. The HEAD and TAIL's style as below:

Data name	Data style	Value	explain
Head	UINT16	0xCCFF	Head
CMD_Size	UINT8		Command length
pCMD	UINT8		Command content
Tail	UINT16	0xFFCC	Tail

Command is in pCMD, CMD_size is the length of Command.

Form documentation

"Data style" is data length. UINT16 is 2 byte. If it’s an integer, data’s sort is big endian. UINT64 is 8 byte. UINT8 is 1 byte.

Style’s definiens is below:

Count		UINT8		Child module data amount
	64_Add	UINT64		Child module 64 bits address
	Out_Add	UINT64		Child module outside number
	Parent_Add	UINT64		Child module 64 bits address
	Status	UINT8	0x00	normal，can pass data
			0x01	Doing initialize network
			0x02	Making sure on-line or not
			0xFF	Condition mistake，need start again
	RFSingle	UINT8		The RSSI when receive RF from parent's module
	DevID	UINT8		Module number

64_Add, Out_Add, Parent_Add, Status, RSSingle, DevID are a data combination. The Count decide amount of combination. Pay attention, when use this command, data length is unfixed. We know the length follow Count.

Status show only 4 values in above style, and every mean of value as explain.

Get version commands

Purpose

This command is for acquire hardware version and software version of module. It tests the applicative command or not. Before use the command, making sure the hardware version and software version fit or not. If not, it will make some unexpected mistakes.

Get_Version (0x13)：

Version data for acquire software.

Applicative module: All.

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x13	
Other_Device	UINT8		Local module or othe module. 0:local;1:Broadcase;3:Network
TagItem_Add	UINT64		Item address

If Other_Device = 1, Coordiantor will broadcast this command for other equipment's response.

If Other_Device = 3, Coordiantor will send this command using network topology.

The item "TagItem_Add" is for Coordiantor only.

If Other_Device = 0 or 1, user can ignore the TagItem_Add.

Get_Version_ACK (0x14)：

Return the information of firmware.

Applicative module: All.

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x14	
Item_Add	UINT64		Item address
Item_PANID	UINT16		Item PANID
MCU_Type	UINT8	0x00	Silconlib
		0x01	PIC
		0x02	Megawin
Version	UINT8	0x01	Firmware version
Behave	UINT8	bit0 = 1	Having FRESH ROM
		bit1 = 1	Coordinator
		bit2 = 1	Router
		bit3 = 1	Device
		bit4 = 1	Remote Setting

ReleaseVersion	UINT16		Release version
WorkVersion	UINT8	0x00	WSN
		0x01	WR

Before using modules, user should check where Version = 0x04 or not.

Fix parameters commands

Purpose

This command is for setting fixed parameter of module. Fixed parameter is effective and wide. If not necessary, we don't execute it. It can avoid some unexpected mistakes.

Set_FixPar (0x17)：

Set fixed parameter of module. This command has tow steps. One step is updating the data into cache memory. Second step is updating cache to FLASH.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x17	
Item_Add	UINT64		64 bits address of Target module
Other_Device	UINT8		Local module or other module. 0:local;3:Network
Action	UINT8		Update target. 0:Cache; 1: From Cache to FLASH
New_Add	UINT64		new 64 bits address

The Firmware uses "Action" to change the 64 bits address on different target.

If Action = 1, down can be ingroed.

Other_Device is only effective in Coordinator. If Other_Device = 3, Coordinator will send this Command to Device base on Item_Add.

After updating the data, it should re-start.

Set_FixPar_ACK (0x18)：

Return the result of setting.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x18	
Item_Add	UINT64		Coordinator address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not

			closely
		0x04	Cache already be used
		0x05	Cache is not FixPar

Get_FixPar (0x19)：

Get fixed parameter of module.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x19	
Item_Add	UINT64		64 bits address of Target module
Other_Device	UINT8		Local module or other module. 0:local;3:Network
Action	UINT8	0x00	Getting target. 0:in Cache; 1:current usage; 2:in FLASH

Other_Device is only effective in Coordinator. If Other_Device = 3, Coordinator will send this Command to Device base on Item_Add.

Get_FixPar_ACK (0x1A)：

Return the result of getting.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1A	
Item_Add	UINT64		64 bits address of Target module
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
		0x04	Cache already be used
		0x05	Cache is not FixPar
Action	UINT8	0x00	Getting target. 0:in Cache; 1:current usage; 2:in FLASH
New_Add	UINT64		64bits address

Zigbee parameters commands

Purpose

Change the parameters of the Zigbee network. These parameters will affect the communication of Zigbee network. Be attention, don't distory the network when changing these parameters.

If you use many Coordinators in the same space, Each Coordinator should have unique value.

Set_Zigbee (0x01)：

Set the Zigbee parameters. This command has tow steps. One step is updating the data into cache memory. Second step is updating cache to FLASH.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x01	
Item_Add	UINT64		Module 64bits address
Other_Device	UINT8		Local module or other module. 0:local;3:Network
Action	UINT8		Update target. 0:Cache; 1: From Cache to FLASH
PANID	UINT16		New PANID
Channel	UINT8		Working Channel
Scan_Channel	UINT8	0x00	Don't scan Channel
		0x01	Scan Channel

The Firmware uses "Action" to change the 64 bits address on different target.

If Action = 1, down can be ingroed.

Other_Device is only effective in Coordinator. If Other_Device = 3, Coordinator will send this Command to Device base on Item_Add.

PANID is the identification of Zigbee network. The Devices which belong to same Network have same PANID. The Channel should be same in the same Zigbee network.

Scan_Channel just work in Device. It decides where Device changes channel automatically when joins the network or not.

Set_Zigbee_ACK (0x02)：

Return the result of setting.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x02	
Item_Add	UINT64		Module 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely
		0x04	Cache already be used
		0x05	Cache is not Zigbee

Get_Zigbee (0x03)：

Get the value of Zigbee parameters.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x03	
Item_Add	UINT64		Module 64bits address
Other_Device	UINT8		Local module or other module. 0:local;3:Network
Action	UINT8	0x00	Getting target. 0:in Cache; 1:current usage; 2:in FLASH

Other_Device is only effective in Coordinator. If Other_Device = 3, Coordinator will send this Command to Device base on Item_Add.

Get_Zigbee_ACK (0x04)：

Return the result of getting.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x04	
Item_Add	UINT64		Module 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
		0x04	Cache already be used
		0x05	Cache is not Zigbee
Action	UINT8	0x00	Getting target. 0:in Cache; 1:current usage; 2:in FLASH
PANID	UINT16		PANID
Channel	UINT8		Working Channel
Scan_Channel	UINT8	0x00	Don't scan Channel
		0x01	Scan Channel

Coordinator commands

Purpose

Setting the parameters of Coodiantor.

Set_Coor (0x09)：

Notice Coordinator to do parameter setting.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x09	
Coor_Add	UINT64		Coordinator 64bits address
RouterType	UINT8	0x00	Routing Table mode
ConfirmType	UINT8	0x00	Adopt PANID identification, receive only the same PANID
		0x01	Adopt 64 bits address identification way
OutsideCheck	UINT8	0x00	Coordinator use inside identification
		0x01	Coordinator use outside identification
MapType	UINT8	0x00	Use 64 bits addrss as obverse number
		0x01	Use outside equipment number as obverse number
ShowLED	UINT8	0x00	Don't turn on On broad LED
		0x01	Turn on On broad LED
LEDType	UINT8	0x00	Flash out wher receive RF single
		0x01	Keep turnon when join network successed
Encryption	UINT8	0x00	Don't use encrytion
		0x01	AES-CTR
		0x02	AES-CCM-128
		0x03	AES-CCM-64
		0x04	AES-CCM-32
		0x05	AES-CBC-MAC-128
		0x06	AES-CBC-MAC-64
		0x07	AES-CBC-MAC-32

ConfirmType sets the identification method when Device joins the network. PANID identification means the Device can join network if PANID of Device is as same as Coordiantor's. User should use command 0x01 and 0x03 to check where the PANID is same or not. Using 64 bits address identification, Coordiantor compare the 64 bit address of Deivce with the data which are stored in the FLASH ROM. User can use command 0x0D to save 64 bits address of Device in FLASH ROM.

OutsideCheck sets which one executes the identification. Inside identification means that Coordinator takes case the indentification. Outside identification means that Coordiantor will pack the device data into command 0x34, then

sending to the other MCU by UART. The other MCU will send the result of indetification using command 0x35.

MapType sets the index is 64 bits address or outside equipment number when finding routing path. If setting MapType = 0x01, the 64 bits address in the comamnd 0x67, 0x62, 0x63 is outside equipment number.

We suggest user use the PANID identification, Coordinator inside identification, 64 bits address maptype.

ShowLED controls where turn on LED which is on the module or not. LEDType decide the working type of LED. The working type is keeping on when on line or flash out when receiving the RF single.

Set_Coor_ACK (0x0A)：

Return the result of setting.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x0A	
Coor_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_Coor (0x0B)：

Getting the parameters of Coordinator

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x0B	
Coor_Add	UINT64		Coordinator 64bits address

Get_Coor_ACK (0x0C)：

Return the parameters of Coordinator.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x0C	
Coor_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not

			closely
RouterType	UINT8	0x00	Routing Table mode
ConfirmType	UINT8	0x00	Adopt PANID identification, receive only the same PANID
		0x01	Adopt 64 bits address identification way
OutsideCheck	UINT8	0x00	Coordinator use inside identification
		0x01	Coordinator use outside identification
MapType	UINT8	0x00	Use 64 bits addrss as obverse number
		0x01	Use outside equipment number as obverse number
ShowLED	UINT8	0x00	Don't turn on On broad LED
		0x01	Turn on On broad LED
LEDType	UINT8	0x00	Flash out wher receive RF single
		0x01	Keep turnon when join network successed
Encryption	UINT8	0x00	Don't use encrytion
		0x01	AES-CTR
		0x02	AES-CCM-128
		0x03	AES-CCM-64
		0x04	AES-CCM-32
		0x05	AES-CBC-MAC-128
		0x06	AES-CBC-MAC-64
		0x07	AES-CBC-MAC-32

Coordinater 64bits_address commands

Purpose

Set 64 bits address of Device that Coordinator can handle. If data amount over UART upper limit, it will dividend several package to send.

Set_64 (0x0D)：

Save 64 bits address of Device in Coordinator.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x0D	
Coor_Add	UINT64		Coordinator 64bits address
Count	UINT8		Amount

	Pos	UINT8		Location of 64 bits address
	64_Add	UINT64		64 bits address of Device

Set_64_ACK (0x0E)：

Return the result of setting.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x0E	
Coor_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_64 (0x0F)：

Getting 64 bits address of Device in Coordinator.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x0F	
Coor_Add	UINT64		Coordinator 64bits address
St_Pos	UINT8		Start location
End_Pos	UINT8		End location

Get_64_ACK (0x10)：

Reply 64 bits address of Device in Coordinator.
Applicative module: Coordinator
Communicate module: Coordinator。

Data name		Data style	Value	Explain
CMD		UINT8	0x10	
Coor_Add		UINT64		Coordinator 64bits address
ACK		UINT8	0x00	Getting OK
			0x01	Format mistake
			0x03	64 bits address not closely
Count		UINT8		Amount
	Pos	UINT8		Location of 64 bits address
	64_Add	UINT64		64 bits address

Get_64_Size (0x11)：

Getting the amount of 64 bits address of Device which be stored in Coordinator.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x11	
Coor_Add	UINT64		Coordinator 64bits address

Get_64_Size_ACK (0x12)：

Retrun the amount of 64 bits address of Device which be stored in Coordinator.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x12	
Coor_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Max_Size	UINT8		The Maximun space
Store_Size	UINT8		usage space

Maximum usage space will change follow different Coordinator.

Del_64 (0x15)：

Delete 64 bits address of Device in Coordinator

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x15	
Coor_Add	UINT64		Coordinator 64bits address
St_Pos	UINT8		Start location
End_Pos	UINT8		End location

St_Pos and End_Pos are both 0, and it means that delete all.

Del_64_ACK (0x16)：

Return the result of delete.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x16	
Coor_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Deleting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Sleep commands

Purpose

Setting the sleep parameters of the system. It can reduce the power usage and extend the life of the battery.

Set_PowerSaving (0x20)：

Setting the parameters
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x20	
Coor_Add	UINT64		Coordinator 64bits address
SleepSend	UINT8		Amount of sending the sleep command
SleepSend_DiffTime	UINT8		The time interval when sending sleep command. Uint: 50ms
Wakeup_Time	UINT8	0x00	Don't Sleep
		0x02	5 seconds
		0x03	10 seconds
		0x04	30 seconds
		0x05	1 minute
		0x06	3 minutes
		0x07	5 minutes
		0x08	10 minutes
Wakeup_Keep	UINT8		Working time. Uint: 50ms
	UINT8	0x00	
	UINT8	0x00	
	UINT8	0x00	
First_Sleep	UINT8		The time what enter sleep mode automatically after power on. Uint: minute

Set_PowerSaving_ACK (0x21)：

Return the result of setting.

Applicative module: Coordinator

Communicate module: Coordinator。

Data name	Data style	Value	Explain
CMD	UINT8	0x21	
Item_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_PowerSaving (0x22)：

Getting the parameters of pwoer saving.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x22	
Coor_Add	UINT64		Coordinator 64bits address

Get_PowerSaving_ACK (0x23)：

Return the result of getting

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x23	
Item_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
SleepSend	UINT8		Amount of sending the sleep command
SleepSend_DiffTime	UINT8		The time interval when sending sleep command. Uint: 50ms
Wakeup_Time	UINT8	0x00	Don't Sleep
		0x02	5 seconds
		0x03	10 seconds
		0x04	30 seconds
		0x05	1 minute
		0x06	3 minutes
		0x07	5 minutes
		0x08	10 minutes
Wakeup_Keep	UINT8		Working time. Uint: 50ms

	UINT8	0x00	
	UINT8	0x00	
	UINT8	0x00	
First_Sleep	UINT8		The time what enter sleep mode automatically after power on. Uint: minute

Device commands

Purpose

Setting the parameters of the Device.

Set_Device (0x05)：

Setting parameters。

Applicative module: Device

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x05	
Device_Add	UINT64		Device 64bits address
LEACHType	UINT8	0x00	Normal Device e
MoveType	UINT8	0x00	Fixed(Router) module
		0x01	Movable(Device) module
Power_mode	UINT8	0x00	Using electric line
		0x01	Using battery
Sleep_Mode	UINT8	0x00	Allow to sleep
		0x01	Don't allow to sleep
ShowLED	UINT8	0x00	Don't turn on On broad LED
		0x01	Turn on On broad LED
LEDType	UINT8	0x00	Flash out wher receive RF single
		0x01	Keep turnon when join network successed
EXTWAKE	UINT8	0x00	Don't use extern wakeup
		0x01	Using extern wakeup
EXTWakeTime	UINT8		Keeping work when be waked up. Unit: Second
Outside_Number	UINT64		

The MoveType can decide where having child or not. The Fixed module can have child. It means Router. The Movable module can't have child. It means End Device.

The Sleep_Mode is the switch of sleep mode. The Fixed module can sleep and the Movable module can't sleep.

ShowLED controls where turn on LED which is on the module or not. LEDType decide the working type of LED. The working type is keeping on when on line or flash out when receiving the RF single.

Set_Device_ACK (0x06)：

Return the result of setting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x06	
Device_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_Device (0x07)：

Getting the parameters of the Device
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x07	
Device_Add	UINT64		Device 64bits address

Get_Device_ACK (0x08)：

Return the result of getting
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x0x08	
Device_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
LEACHType	UINT8	0x00	Normal Device e
MoveType	UINT8	0x00	Fixed(Router) module
		0x01	Movable (Device) module
Power_mode	UINT8	0x00	Using electric line
		0x01	Using battery
Sleep_Mode	UINT8	0x00	Allow to sleep
		0x01	Don't allow to sleep
ShowLED	UINT8	0x00	Don't turn on On broad LED
		0x01	Turn on On broad LED
LEDType	UINT8	0x00	Flash out wher receive RF single
		0x01	Keep turnon when join network successed

EXTWAKE	UINT8	0x00	Don't use extern wakeup
		0x01	Using extern wakeup
EXTWakeTime	UINT8		Keeping work when be waked up. Unit: Second
Outside_Number	UINT64		

Setting network commands

Purpos

Set parameter of network, and it builds network topology and the quality of the network.

Set_Netwok (0x1B)：

Setting the parameters of network.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1B	
Item_Add	UINT64		Device 64bits address
Auto_Parent	UINT8	bit 0	Use default Parent first then find Parent automatically 0: Yes. 1: No.
		bit 1	Only use default Parent or not. 0:No; 1:Yes
Parent_Check	UINT 8		The time period to make sure that Parent is alive.Unit is second, minimum:2
MinRSSI	UINT8		minimal intensity of acceptable RSSI
Parent_Count	UINT8		Default Parent amount
Parent_Add	UINT64		Default Parent's 64its address

If bit 0 of Auto_Parent = 1, Device finds a network by broadcast, the bit 0 of Auto_Parent = 0 Device finds a network by 64 bits address. 64 bits address of goal saves in Parent_Add.

If bit 1 of Auto_Parent = 1, Device only finds the network by 64 bits address. So Device can't fix network automatically.

Parent_Check means how long the module checks the network survival. If the network isn’t survival, it would search network again. If this parameter set too small, it makes RF package of network are all survival detect. It reduces the speed of pass. Beyond setting makes module off-line too long. It modifies more follow actual demand.

MinRSSI is the minimal value to receive RF message, and beyond this RF

message of value to deal. It makes sure the quality of receiving RF message.
Parent_Count's maximum amount is 2.

Set_Network_ACK (0x1C)：

Return the result of setting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1C	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_Network (0x1D)：

Getting the parameters of Network.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1D	
Item_Add	UINT64		Device 64bits address

Get_Network_ACK (0x1E)：

Return the result of getting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1E	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Auto_Parent	UINT8	bit 0	Only use default Parent or not, and don't find actively. 0 : use default. 1 : find actively.
Parent_Check	UINT8		The time period to make sure that Parent is alive.Unit is second, minimum:2
MinRSSI	UINT8		minimal intensity of acceptable RSSI
Parent_Count	UINT8		Default Parent amount
Parent_Add	UINT64		Default Parent's 64its

				address

Setting network commands

Purpos

Set parameter of network, and it builds network topology and the quality of the network.

Set_Netwok (0x1B)：

Setting the parameters of network.

Applicative module: Device

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1B	
Item_Add	UINT64		Device 64bits address
Auto_Parent	UINT8	bit 0	Use default Parent first then find Parent automatically 0: Yes. 1: No.
		bit 1	Only use default Parent or not. 0:No; 1:Yes
Parent_Check	UINT 8		The time period to make sure that Parent is alive.Unit is second, minimum:2
MinRSSI	UINT8		minimal intensity of acceptable RSSI
Parent_Count	UINT8		Default Parent amount
Parent_Add	UINT64		Default Parent's 64its address

If bit 0 of Auto_Parent = 1, Device finds a network by broadcast, the bit 0 of Auto_Parent = 0 Device finds a network by 64 bits address. 64 bits address of goal saves in Parent_Add.

If bit 1 of Auto_Parent = 1, Device only finds the network by 64 bits address. So Device can't fix network automatically.

Parent_Check means how long the module checks the network survival. If the network isn’t survival, it would search network again. If this parameter set too small, it makes RF package of network are all survival detect. It reduces the speed of pass. Beyond setting makes module off-line too long. It modifies more follow actual demand.

MinRSSI is the minimal value to receive RF message, and beyond this RF message of value to deal. It makes sure the quality of receiving RF message.

Parent_Count’s maximum amount is 2.

Set_Network_ACK (0x1C)：

Return the result of setting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1C	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_Network (0x1D)：

Getting the parameters of Network.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1D	
Item_Add	UINT64		Device 64bits address

Get_Network_ACK (0x1E)：

Return the result of getting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x1E	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Auto_Parent	UINT8	bit 0	Only use default Parent or not, and don't find actively. 0 : use default. 1 : find actively.
Parent_Check	UINT8		The time period to make sure that Parent is alive.Unit is second, minimum:2
MinRSSI	UINT8		minimal intensity of acceptable RSSI
Parent_Count	UINT8		Default Parent amount
Parent_Add	UINT64		Default Parent's 64its address

Baud rate commands

Purpos

Changing the baud rate of Device RS232

Set_UR (0x24)：

Setting the baud rate.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x24	
Item_Add	UINT64		Device 64bits address
Action	UINT8	0x00	Update FLASH and Work immediately
		0x01	Update FLASH only
Trans	UINT8	0x00	Using QR format
		0x01	Using transparent mode
BaudRate	UINT8	0x01	1200 bps
		0x02	2400 bps
		0x03	9600 bps
		0x04	14400 bps
		0x05	19200 bps
		0x06	38400 bps
		0x07	57600 bps
		0x08	115200 bps
Parity_Check	UINT8	0	None Check
		1	Even Check
		2	Odd Check

Set_UR_ACK(0x25)：

Return the result of setting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x25	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Setting OK
		0x06	Baud rate unknow
		0x07	Parity checj unknow
		0x03	64 bits address not closely

Get_UR (0x26)：

Getting the baud rate of the Device.
Applicative module: Device

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x26	
Item_Add	UINT64		Device 64bits address
Action	UINT8	0x00	Current usage
		0x01	in FLASH

Get_UR_ACK(0x27)：

Return the result of getting.

Applicative module: Device

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x27	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Trans	UINT8	0x00	Using QR format
		0x01	Using transparent mode
BaudRate	UINT8	0x01	1200 bps
		0x02	2400 bps
		0x03	9600 bps
		0x04	14400 bps
		0x05	19200 bps
		0x06	38400 bps
		0x07	57600 bps
		0x08	115200 bps
Parity_Check	UINT8	0	None Check
		1	Even Check
		2	Odd Check

Other commands

Purpos

Setting parameter.

Set_Other (0x28)：

Setting the parameters.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x28	
Item_Add	UINT64		Moudle 64bits address
ModuleType	UINT8	0x00	For Coordinator
		0x01	For Device

WorkMode	UINT 8	0x00	Detecting mode
		0x01	WSN mode
Low_Power	UINT8	0x00	0db
		0x28	-3.7db
		0x38	-6.3db
		0x40	-10db
		0x68	-13.7db
		0x78	-16.3db
		0x80	-20db
		0xA8	-23.7db
		0xB8	-26.3db
		0xC0	-30db
		0xE8	-33.7db
		0xF8	-36.3db
ConfirmMode	UINT8	0x00	Coordinator no-confirm mode
		0x01	Coordinator confirm mode
WakeLength	UINT8		Unit : ms
Append2	UINT8	0	
Append3	UINT8	0	
Append4	UINT8	0	
Append5	UINT8	0	

If ModuleType = 1 and this command be sended to Cooridnator, Coordiantor will re-send to Device depend on the Item_Add.

Low_Power will reduce the power of RF single.

The ConfirmMode is for Coordinator only.

The WakeLength decides the length of single which wakes up the outside MCU.

Set_Other_ACK (0x29)：

Return the result of setting.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x29	
Item_Add	UINT64		Module 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_Other (0x2A)：

Getting the parameters.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain

CMD	UINT8	0x2A	
Item_Add	UINT64		Module 64bits address
ModuleType	UINT8	0x00	For Coordinator
		0x01	For Device

Get_Other_ACK (0x2B)：

Return the result of getting.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x2B	
Item_Add	UINT64		Module 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
ModuleType	UINT8	0x00	For Coordinator
		0x01	For Device
WorkMode	UINT 8	0x00	Detecting mode
		0x01	WSN mode
Low_Power	UINT8	0x00	0db
		0x28	-3.7db
		0x38	-6.3db
		0x40	-10db
		0x68	-13.7db
		0x78	-16.3db
		0x80	-20db
		0xA8	-23.7db
		0xB8	-26.3db
		0xC0	-30db
		0xE8	-33.7db
		0xF8	-36.3db
ConfirmMode	UINT8	0x00	Coordinator no-confirm mode
		0x01	Coordinator confirm mode
WakeLength	UINT8		Unit : ms
Append2	UINT8	0	
Append3	UINT8	0	
Append4	UINT8	0	
Append5	UINT8	0	

Sending data commands

Purpos

Packing the outside data and pass out in the network

BCRaw_Data (0x66)：

This command is maked from outside MCU of Coordinator. Sending user-define data pass all modules by broadcast, the module pass this command to his outside MCU again.

Applicative module: All

Communicate module: All

Data name		Data style	Value	Explain
CMD		UINT8	0x66	
SrcMapAddress		UINT64		Number of source
DataSize		UINT8		Data length
	pData	UINT8		data

SrcMapAddress fills 0x00. DataSize maximum is 60.

Raw_Data (0x67)：

This command is maked from outside MCU of Coordinator or Device. Sending user-define data pass to one module by one module, the module pass this command to his outside MCU again.

Applicative module: All

Communicate module: All

Data name		Data style	Value	Explain
CMD		UINT8	0x67	
DestMapAddress		UINT64		Number of destination
SrcMapAddress		UINT64		Number of source
DataSize		UINT8		Data length
	pData	UINT8		Data

We only write DestMapAddress in sending command, and send to module. Then the module adds SrcMapAddress and passess to the goal. When we response the comand, we must take out number of goal from received command (it means the SrcMapAddress). After dealing the command, we write this value in column of DestMapAddress.

Device must pass command to Coordinator. After Coordinator receives it, deciding passing to outside MCU (DestMapAddress are all 0xFF) or the other module base on the DestMapAddress.

DataSize's maximum is 60

Raw_Data_Send (0x69)：

This command is the response of command 0x67. It means data already be send out by RF.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x69	

System poerator commands

Purpose

To provide user command that can operate system. It's for change the state of system, and makes system more stable.

System_Status (0x70)：

Getting the status of the module.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x70	
Item_Add	UINT64		Module 64bits address
Other_Device	UINT8		Local module or other module. 0:local;1:other

Other_Device is only effective in Coordinator.

System_Status_ACK (0x71)：

Return the status of Device.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x71	
Item_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x02	Don't support this command
Item_Status	UINT8	0x00	normal, can pass data
		0x01	Doing initialize network
		0x02	Making sure on-line or not
		0x03	Connect fail
		0xFF	Status mistake, need start again
Parent_64Add	UINT64		Parent's 64bits address
RFSingle	UINT8		The RSSI when receive RF from Parent module
DevID	UINT8		Device number
Power_Status	UINT8	bit 0	0: using electric line; 1: usage battery
		bit 1	0: Allow to sleep; 1: Don't allow to sleep

Return the status of Coordinator
Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x71	
Item_Add	UINT64		Coordinator 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x02	Don't support this command
Item_Status	UINT8	0x00	normal, can pass data
		0x01	Doing initialize network
		0x02	Making sure on-line or not
		0x03	Connect fail
		0xFF	Status mistake, need start again
Sleep_Control	UINT8	0x00	Don't Allow to sleep
		0x01	Allow to sleep

System_Restart (0x72)：

Restart the module

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x72	
Item_Add	UINT64		Module 64bits address
Other_Device	UINT8		Local module or other module. 0:local;1:other

Other_Device is only effective in Coordinator. If Other_Device = 1, Coordinator will send this Command to Device base on Item_Add.

Get_Child (0x73)：

Get the 64 bits address of child module in the meomry of Coordinator.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x73	
Coor_Add	UINT64		Coordinator 64bits address
St_Pos	UINT8		Start location
End_Pos	UINT8		End location

The definition of child module is that has been entered the network successfully.

Before use this command, using command 0x76 to acquire maximum rang of child module.

Get_Child_ACK (0x74)：

Return the 64 bits address of child module.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x74	
Coor_Add	UINT64		Coordinator 64ibts address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Count	UINT8		Amount
64_Add	UINT64		Child's 64 bits address

Because UART has limit space, Coordiantor passes the one child data out one time. Use command 0x76 to make sure all 64 bits address of child module pass out or not.

Get_Child_Size (0x75)：

Get the amount of child module in the meomry of Coordinator.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x75	
Coor_Add	UINT64		Coordinator 64bits address

Get_Child_Size_ACK (0x76)：

Return the amount of on-line child module and possible maximum amount of child module.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x76	
Coor_Add	UINT64		Coordinator 64 bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Max_Size	UINT8		Maximum child module amount
Store_Size	UINT8		Online child module amount

Get_Child_Data (0x77)：

Getting the data of child modules in the meomry of Coordiantor.
Applicative module: Coordinator
Communicate module: Coordinator

Data name		Data style	Value	Explain
CMD		UINT8	0x77	
Coor_Add		UINT64		Coordinator address
Count		UINT8		Child module amount
	64_Add	UINT64		Child module 64 bits address

Count = 0 means that pass all Child module.
Whole data doesn't beyond 83bytes.

Get_Child_Data_ACK (0x78)：

Return the datas of child module.
Applicative module: Coordinator
Communicate module: Coordinator

Data name		Data style	Value	Explain
CMD		UINT8	0x78	
Coor_Add		UINT64		Coordinator 64bits address
ACK		UINT8	0x00	Setting OK
			0x01	Format mistake
			0x03	64 bits address not closely
Count		UINT8		Amount of child module
	64_Add	UINT64		64 bits address of child module
	Out_Add	UINT64		Outside number of child module
	Parent_Add	UINT64		64 bits address of Parent's moudme
	Status	UINT8	0x00	OK
			0x01	Runing initialize network
			0x02	Make sure where on-line or not
			0xFF	Condition mistake，need start again
	RFSingle	UINT8		The RSSI when receive RF from parent's module
	DevID	UINT8		Module number
	Power_Status	UINT8	bit 0	0: using electric line; 1: usage battery
			bit 1	0: Allow to sleep; 1: Don't allow to sleep

Because UART is limit space, Coordiantor would pass child data one by one. You can use command 0x76 to make sure that all child data pass out or not.

If you need data that isn't in response, you would use other command.

System_Reboot (0x79)：

Re-start all modules in the system.

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x79	
Item_Add	UINT64		Coordinator 64bits address

After Coordinator receives, it would pass command to Device to start again by broadcast.

Get_Item_Data(0x84)：

Getting the 64 bits addrrss of Coordiantor.

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x84	

Get_Item_Data_ACK(0x85)：

回應取得模組與 Coordinator 的 64 bits addrrss。

Applicative module: All

Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x85	
Status	UINT8	0x00	Not yet join the network
		0x01	Join the network successed
Item_Add	UINT64		Module 64bits address
Coor_Add	UINT64		Coordinator 64bits address

Check_Child_Alive (0x86)：

Coordinator detect where all Devices are alive or not

Applicative module: Coordinator

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x86	
Item_Add	UINT64		Coordiantor 64bits address

Coordiantor ask all Devices to report the routing table back.

Check_Child_Alive_ACK (0x87)：

Notify outside MCU that Cooridnator executes the command 0x86
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x87	
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Action	UINT8	0x00	Can't start
		0x01	Start checking

Ping (0x88)：

Ask the Moudle back a single.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x88	
Item_Add	UINT64		Module 64bits address
Data	UINT16		
SendType	UINT8	0x00	Using broadcast
		0x01	Sending to Item_Add
			Using broadcast and filter the return command by Item_Add

The Data is the identification number. Device will send it back. So outside MCU can check.

If SendType = 0x00, the Module sends this command using broadcast way. It SendType = 0x02, the Module sends this command using broadcast and filters the return message by Item_Add.

Ping_ACK (0x89)：

Reutrn vlaue of Module.
Applicative module: All
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0x89	
Item_Add	UINT64		Device 64bits address
Data	UINT16		
Parent_Add	UINT64		64bits address of Parent
RSSI	UINT8		RSSI when receiving command 0x88

In WSN mode, the Parent_Address is the real address of Parent. In Detect mode, the

Parent_Address is the Address which send the command 0x88.

Ask_Wakeup (0x8A)：

Wake up some module to work.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x8A	
Coor_Add	UINT64		Coordinator 64bits address
WakeTime	UINT8		Wake up time. Unit: second
Item_Address	UINT64		64bit address of the module which will be waked up

If Item_Address = 0xFFFFFFFFFFFFFFFF, it means that wake up all modules.

Ask_Wakeup_ACK (0x8B)：

Notify outside MCU that some module is working now.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x8B	
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Action	UINT8	0x00	Action finish
		0x01	Module reports result
Item_Address	UINT64		64 bit address of Device which be waked up

If Action = 0x00, we can ignore the Item_Address.

Sleep_Control (0x8C)：

Setting Network is in the sleep status or work status.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x8C	
Coor_Add	UINT64		Coordiantor 64bits address
Action	UINT8	0x00	Start sleep
		0x01	Srop sleep

Sleep_Control_ACK (0x8D)：

Return the result of command Sleep_Control。
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x8D	
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

CurrentTime (0x8F)：

Setting the GMT time of Coordiantor.
Applicative module: Coordinator
Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x8F	
Coor_Add	UINT64		Coordiantor 64bits address
Action	UINT8	0x00	Setting GMT time
		0x01	Getting GMT time
Year	UINT8		Year subtract 2000
Month	UINT8		Month(0-11)
Day	UINT8		Day(0-30)
Hour	UINT8		Hour(0-23)
Minute	UINT8		Minute(0-59)
Second	UINT8		Second(0-59)

Setting Sensor commands

Purpose

Setting the parameters of the Sensors. It makes a standard how to deal with the sensor.

Set_Sensor (0xB0)：

Setting the parameters of the Sensors and the frequency of reading sensor data.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0xB0	
Device_Add	UINT64		Device 64bits address
DeviceType	UINT8	0x00	Sensor Device
		0x01	RAW Device

Data_Send_Time	UINT16		The time period when Device send data out actively. 0: doesn't send back. Uint : ms
SensorType	UINT8	0x00	CO Sensor
		0x01	Gas Sensor
		0x02	Temp. & RH Sensor
		0x03	Somg Sensor
		0x04	Move Sensor
		0x05	Switch control
		0x06	Other Sensor
		0x07	Light control
		0x08	V input
		0x09	I input
SensorPartNum	UINT8		Sensor type number

For stable network and less RF package collision probability, propose the user-define Data_Send_Time bigger then 500ms.

SensorType and SensorPartNum idenyify distinguish what kind of the Sensor that Device connects. User can change different Sensor follow plan of system.

Set_Sensor_ACK (0xB1)：

Return the result of setting.
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0xB1	
Device_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

Get_Sensor (0xB2)：

取得 Device 目前的 Sensor 設定。
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0xB2	
Device_Add	UINT64		Device 64bits address

Get_Sensor_ACK (0xB3)：

回覆的 Sensor 的設定值。
Applicative module: Device
Communicate module: All

Data name	Data style	Value	Explain
CMD	UINT8	0xB3	
Device_Add	UINT64		Device 64bits address
ACK	UINT8	0x00	Getting OK
		0x01	Format mistake
		0x03	64 bits address not closely
DeviceType	UINT8	0x00	Sensor Device
		0x01	RAW Device
Data_Send_Time	UINT16		The time period when Device send data out actively. 0: doesn't send back. Uint : ms
SensorType	UINT8	0x00	CO Sensor
		0x01	Gas Sensor
		0x02	Temp. & RH Sensor
		0x03	Somg Sensor
		0x04	Move Sensor
		0x05	Switch control
		0x06	Other Sensor
		0x07	Light control
		0x08	V input
		0x09	I input
SensorPartNum	UINT8		Sensor type number

Sensor data transform commands

Purpose

Sensor Device send sensor data to Coordinator。

Sensor_Data (0x62)：

This command is maked by Device. Device passes this command to Coordinator, then pass to outside MCU.

Applicative module: Device

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x62	
Device_Address	UINT64		Device 64bits address
SData_Type	UINT8	0	Type of Sensor data
INPUT	UINT8	bit 0	0 : Digital input is low ; 1 : Digital input is high
ANA	UINT16		Anagle input

Because ON and OFF of digital input bases on outside PCB layout, Device can't decide ON or OFF when it read single. Just pass the state (High potential or Low potential) to Coordiantor, and Coordiantor decides ON/OFF.

Sensor_Data_ACK (0x63)：

This command is maked by Outside MCU. Passing this command to the Device to change the outputs by Coordinator.

Applicative module: Device

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x63	
Device_Address	UINT64		Device 64bits address
SData_Type	UINT8	0	Type of Sensor data
OUTPUT	UINT8	bit 0	0:Digital output is low; 1:Digital output is high

Because ON and OFF of digital output bases on outside PCB layout, Device can't decide ON or OFF using command. Just change the state (High potential or Low potential) of pin.

GetSensorData (0x64)：

Getting sendor data

Applicative module: Device

Communicate module: Coordinator

Data name	Data style	Value	Explain
CMD	UINT8	0x64	
Device_Address	UINT64		Device 64bits address

Device returns the command 0x62.

Getting child sensor data commands

Purpose

Getting the Sensor data of child module is the memory of Coordinator.

Get_Child_Sensor_Data (0xB4)：

Get on-line sensor data in the Coordinator.

Applicative module: Coordinator

Communicate module: Coordinator

Data name		Data style	Value	Explain
CMD		UINT8	0xB4	
Coor_Add		UINT64		Coordinator 64bits address
Count		UINT8		Amount of child module
	64_Add	UINT64		64 bits address of child module

Count = 0 means pass all Child.

Whole data's size can't beyond 83bytes.

Get_Child_Sensor_Data_ACK (0xB5)：

Return sensor data of child module.
Applicative module: Coordinator
Communicate module: Coordinator

Data name		Data style	Value	Explain
CMD		UINT8	0xB5	
Coor_Add		UINT64		Coordinator 64bits address
ACK		UNIT8	0x00	Getting OK
			0x01	Format mistake
			0x03	64 bits address not closely
Count		UINT8		Amount of child module
	64_Add	UINT64		64 bits address of child module
	Data_Send_Time	UINT16		The time period when Device send data out actively. Uint : ms
	SensorType	UINT8	0x00	CO Sensor
			0x01	Gas Sensor
			0x02	Temp. & RH Sensor
			0x03	Somg Sensor
			0x04	Move Sensor
			0x05	Switch control
			0x06	Other Sensor
			0x07	Light control
			0x08	V input
			0x09	I input
	SensorPartNum	UINT8		Sensor type number

Because UART is limit space, Coordiantor would pass child sensor data one by one. You can usecommand 0x76 to make sure that all child data pass out or not.

If you need data that isn't in response, you would use other command.

5. Sample of commands

Noun explain

Program: The application program which user uses can run in PC of Embeddedsystem. It can be Coor program: work with Coorinator, Device program: work with Device.

Module: include Coordinator or Device.

Module identification

The first function which the program should run is module identifincation. The purpose is that the program should identify the type and the Frimware version of the modules. If it finds thar the version fo the module doesn't match, it should stops. The command of module identification is 0x13, 0x14.

Commands

Program sends: 0xCC 0xFF 0x02 0x13 0x00 0xFF 0xCC

Module retuens: 0xCC 0xFF 0x11 0x14 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x51 0x52 0x04 0x09 0x20 0x35 0x00 0xFF 0xCC

Return analysis

64bits address of the module: 5152540000000710

PANID of the module: 5152

Command version: 04

Type of the module: 09, Device

Version of the Firmware: 204

Hardware: 5, QRZ-3100

There are tow points that uses should notice. One is that the Command version is 04, the other is that the version of the firmware is 204.

Default parameters

The modules will be setted the default parameters on the market. So user can build a network simply and send data

The commands for default parameters on the Coordinator side are 0x01, 0x09, 0x20, on the Device side are 0x01, 0x05, 0x1B, 0x24, 0xB0,

The defaulf parameters of Coordinator:

Command 0x01

Data name	Data style	Value	Explain
CMD	UINT8	0x01	
Item_Add	UINT64		Coordinator 64bits

			address
Other_Device	UINT8	0x00	
Action	UINT8		
PANID	UINT16	0x5152	New PANID
Channel	UINT8	0x04	

Command 0x09

Data name	Data style	Value	Explain
CMD	UINT8	0x09	
Coor_Add	UINT64		Coordinator 64bits address
RouterType	UINT8	0x00	Routing Table mode
ConfirmType	UINT8	0x00	Adopt PANID identification, receive only the same PANID
OutsideCheck	UINT8	0x00	Coordinator use inside identification
MapType	UINT8	0x00	Use 64 bits addrss as obverse number
ShowLED	UINT8	0x01	Turn on On broad LED
LEDType	UINT8	0x01	Keep turnon when join network successed
Encryption	UINT8	0x00	Don't use encrytion

Comand 0x20

Data name	Data style	Value	Explain
CMD	UINT8	0x20	
Coor_Add	UINT64		Coordinator 64bits address
SleepSend	UINT8	0x02	Amount of sending the sleep command
SleepSend_DiffTime	UINT8	0x02	The time interval when sending sleep command. Uint: 50ms
Wakeup_Time	UINT8	0x00	Don't Sleep
Wakeup_Keep	UINT8	0x01	Working time. Uint: 50ms
	UINT8	0x00	
	UINT8	0x00	
	UINT8	0x00	
First_Sleep	UINT8	0x0A	The time what enter sleep mode automatically after power on. Uint: minute

The default parameters of Device:

Command 0x01

Data name	Data style	Value	Explain
CMD	UINT8	0x01	
Item_Add	UINT64		Device 64bits address
Other_Device	UINT8	0x00	
Action	UINT8		
PANID	UINT16	0x5152	New PANID
Channel	UINT8	0x04	
Scan_Channel	UINT8	0x00	Don't scan Channel

Command 0x05

Data name	Data style	Value	Explain
CMD	UINT8	0x05	
Device_Add	UINT64		Device 64bits address
LEACHType	UINT8	0x00	Normal Device e
MoveType	UINT8	0x00	Fixed(Router) module
Power_mode	UINT8	0x01	Using battery
Sleep_Mode	UINT8	0x01	Don't allow to sleep
ShowLED	UINT8	0x01	Turn on On broad LED
LEDType	UINT8	0x01	Keep turnon when join network successed
EXTWAKE	UINT8	0x00	Don't use extern wakeup
EXTWakeTime	UINT8	0x01	Keeping work when be waked up. Unit: Second
Outside_Number	UINT64		0000000000000000

Comamnd 0x1B

Data name	Data style	Value	Explain
CMD	UINT8	0x1B	
Item_Add	UINT64		Device 64bits address
Auto_Parent	UINT8	0x01	Only use default Parent or not, and don’t find actively. 0 : use default. 1 : find actively.
Parent_Check	UINT 8	0x05	The time period to make sure that Parent is alive.Unit is second, minimum:2
MinRSSI	UINT8	0x00	minimal intensity of acceptable RSSI
Parent_Count	UINT8	0x01	Default Parent amount
Parent_Add	UINT64		0000000000000000

Command 0x24

Data name	Data style	Value	Explain
CMD	UINT8	0x24	
Item_Add	UINT64		Device 64bits address

Action	UINT8		
Trans	UINT8	0x00	Using QR format
BaudRate	UINT8	0x03	9600
Parity_Check	UINT8	0	None Check

Command 0xB0

Data name	Data style	Value	Explain
CMD	UINT8	0xB0	
Device_Add	UINT64		Device 64bits address
DeviceType	UINT8	0x01	RAW Device
Data_Send_Time	UINT16	0x03E8	The time period when Device send data out actively. Uint : ms
SensorType	UINT8	0x00	CO Sensor
SensorPartNum	UINT8	0x00	Sensor type number

Commands

The sample of Coor program sends(64bits addess of Coodinator = 5152544300000073):

0xCC 0xFF 0x10 0x09 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x00 0x00 0x00 0x00 0x01 0x01 0x00 0xFF 0xCC

0xCC 0xFF 0x11 0x20 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x02 0x02 0x00 0x01 0x00 0x00 0x00 0x0A 0xFF 0xCC

0xCC 0xFF 0x0E 0x01 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x00 0x00 0x51 0x52 0x04 0xFF 0xCC

0xCC 0xFF 0x0E 0x01 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x00 0x01 0x51 0x52 0x04 0xFF 0xCC

The sample of Device program sends(64bits addess of Device = 5152544300000073):

0xCC 0xFF 0x19 0x05 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x00 0x00 0x01 0x01 0x01 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0xCC

0xCC 0xFF 0x15 0x1B 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x01 0x05 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0xCC

0xCC 0xFF 0x0D 0x24 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x01 0x00 0x03 0x00 0xFF 0xCC

0xCC 0xFF 0x0E 0xB0 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x01 0x03 0xE8 0x00 0x00 0xFF 0xCC

0xCC 0xFF 0x0F 0x01 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x00 0x51 0x52 0x04 0x00 0xFF 0xCC

0xCC 0xFF 0x0F 0x01 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x01 0x51 0x52 0x04 0x00 0xFF 0xCC

Using QR-format to send data

User can pack the data into the command 0x67 and sends the other modules. This is QR-format. The maximum payload of the command 0x67 is 60. If the length of data is bigger than 60, it should be sent by several commands.

Samlpes

The 64bits address of Coordinator is 5152544300000073, the 64bits addess of Device is 5152540000000710, the data is 0x1234

Coordinator sends to Deivce:
0xCC 0xFF 0x14 0x67 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x02 0x12 0x34 0xFF 0xCC

Device sends to Coordinator:
0xCC 0xFF 0x14 0x67 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x02 0x12 0x34 0xFF 0xCC
or
0xCC 0xFF 0x14 0x67 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x02 0x12 0x34 0xFF 0xCC

Using transparent to send data

When you have fixed communication potocol and can't change it on the Device side, you can use the transparent format. Then you can use the Device program without any change.

If Device uses tramsparent mode, the parameters of the Device can't be change by UR. User should RF to change the parameters of the Device.

The command about transparent is 0x24.

Command 0x24

Data name	Data style	Value	Explain
CMD	UINT8	0x24	
Item_Add	UINT64		Device address
Action	**UINT8**	**0x00**	**Update FLASH and Work immediately**
Trans	**UINT8**	**0x01**	**Using transparent mode**
BaudRate	UINT8	0x03	9600
Parity_Check	UINT8	0	None Check

The Action can work immedinately. The Trans equals 0x01 means using the transparent mode.

You still use the QR-format on Coordinator side. At that time, the length of payload in command 0x67 is **40**.

Samples

The 64bits address of Coordinator is 5152544300000073, the 64bits addess of Device is 5152540000000710, the data is 0x1234

Chaging to transparent mode:
0xCC 0xFF 0x0D 0x24 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x01 0x03 0x00 0xFF 0xCC

Coordinator sends to Deivce：
0xCC 0xFF 0x14 0x67 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x02 0x12 0x34 0xFF 0xCC

Device sends to Coordinator:
0x12 0x34

Sensor data

When Device is setted as Sensor Device, it will read GPIO and send out using comand 0x62 automatically.
Thg command about Sensor Device is 0xB0.

Command 0xB0

Data name	Data style	Value	Explain
CMD	UINT8	0xB0	
Device_Add	UINT64		Device address
DeviceType	**UINT8**	**0x00**	**Sensor Device**
Data_Send_Time	UINT16	0x03E8	The time period when Device send data out actively. Uint : ms
SensorType	UINT8	0x00	CO Sensor
SensorPartNum	UINT8	0x00	Sensor type number

Command 0x62

Data name	Data style	Value	Explain
CMD	UINT8	0x062	
Device_Address	UINT64		Device 64bits address
SData_Type	UINT8	0	Type of Sensor data
INPUT	UINT8	bit 0	0 : Digital input is low ; 1 : Digital input is high
ANA	UINT16		Anagle input

Samples

Device program sends(64bits addess of Device = 5152540000000710)：
0xCC 0xFF 0x0E 0xB0 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x03 0xE8 0x00 0x00 0xFF 0xCC

Coordinator program receives
0xCC 0xFF 0x0D 0x62 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x00 0x00 0xE0 0xFF 0xCC

Sleep

There are two sets of sleep commands, one is setting parameters (commands are 0x20, 0x05), the other is control(commands are 0x08A, 0x8B,0x8C,0x8D,0x8F).

Command 0x20 sets the sleep parameters on Coordinator side. The Wakeup_Time sets that Device wakes up every 10 seconds, The Wakeup_Keep sets that Device keeps awake 100ms.

Data name	Data style	Value	Explain
CMD	UINT8	0x20	
Coor_Add	UINT64		Coordinator address
SleepSend	UINT8	0x02	Amount of sending the sleep command
SleepSend_DiffTime	UINT8	0x02	The time interval when sending sleep command. Uint: 50ms
Wakeup_Time	**UINT8**	**0x03**	**10 seconad**
Wakeup_Keep	**UINT8**	**0x02**	**Working time. Uint: 50ms**
	UINT8	0x00	
	UINT8	0x00	
	UINT8	0x00	
First_Sleep	UINT8	0x0A	The time what enter sleep mode automatically after power on. Uint: minute

Command 0x05 sets the sleep commands on Device side. The MoveType equals 0x01, the Sleep_Mode equals 0x00.

Data name	Data style	Value	Explain
CMD	UINT8	0x05	
Device_Add	UINT64		Device address
LEACHType	UINT8	0x00	一般的 Device
MoveType	**UINT8**	**0x01**	**Movable(Device) module**
Power_mode	**UINT8**	**0x01**	**Using battery**
Sleep_Mode	**UINT8**	**0x00**	**Allow to sleep**
ShowLED	UINT8	0x01	Turn on On broad LED
LEDType	UINT8	0x01	Keep turnon when join network successed
EXTWAKE	**UINT8**	**0x00**	**Don't use extern**

			wakeup
EXTWakeTime	**UINT8**	**0x01**	**Keeping work when be waked up. Unit: Second**
Outside_Number	UINT64		0000000000000000

Cooridnator can wake some module or all modules up using command 0x8A and 0x8B

Command 0x8A

Data name	Data style	Value	Explain
CMD	UINT8	0x8A	
Coor_Add	UINT64		Coordinator 64bits address
WakeTime	**UINT8**		**Wake up time. Unit: second**
Item_Address	**UINT64**		**64bit address of the module which will be waked up**

Command 0x8B

Data name	Data style	Value	Explain
CMD	UINT8	0x8B	
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely
Action	**UINT8**	**0x00**	**Action finish**
		0x01	**Module reports result**
Item_Address	UINT64		64 bit address of Device which be waked up

The Coordinator starts or stops the sleep using command 0x8C and 0x8D.

Command 0x8C

Data name	Data style	Value	Explain
CMD	UINT8	0x8C	
Coor_Add	UINT64		Coordiantor 64bits address
Action	**UINT8**	**0x00**	**Start sleep**
		0x01	**Srop sleep**

Command 0x8C

Data name	Data style	Value	Explain
CMD	UINT8	0x8D	
ACK	UINT8	0x00	Setting OK
		0x01	Format mistake
		0x03	64 bits address not closely

If the Device is waked, it will module send the command 0x8E to the Device porgram.

Command 0x8E

Data name	Data style	Value	Explain
CMD	UINT8	0x8E	

Samlpes

Coor program sends(64bits addess of Coordinator= 5152544300000073)：

0xCC 0xFF 0x11 0x20 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x02 0x02 0x03 0x02 0x00 0x00 0x00 0x0A 0xFF 0xCC

Device program sends(64bits addess of Device = 5152540000000710)：

0xCC 0xFF 0x19 0x05 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x00 0x01 0x01 0x00 0x01 0x01 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0xCC

Coor program wakes one Device up:

0xCC 0xFF 0x12 0x8A 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x02 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0xFF 0xCC

Detecting modules

When user installs the moduels, she must know the quality of RF single. If the quailty is bad, user could get big data lost rate. So user can use detecting mode to send command 0x88 to nearby modules and get the quailty using RSSI.

Coordinator Side

Command 0x28

Data name	Data style	Value	Explain
CMD	UINT8	0x28	
Item_Add	UINT64		Module 64bits address
ModuleType	**UINT8**	**0x00**	**For Coordinator**
WorkMode	**UINT8**	**0x00**	**Detecting mode**
Low_Power	UINT8	0x00	0db
ConfirmMode	UINT8	0x01	
Append1	UINT8	0	
Append2	UINT8	0	
Append3	UINT8	0	
Append4	UINT8	0	
Append5	UINT8	0	

Device side

Command0x28

Data name	Data style	Value	Explain

CMD	UINT8	0x28	
Item_Add	UINT64		Module 64bits address
ModuleType	**UINT8**	**0x01**	**For Device**
WorkMode	**UINT8**	**0x00**	**Detecting mode**
Low_Power	UINT8	0x00	0db
ConfirmMode	UINT8	0x01	
Append1	UINT8	0	
Append2	UINT8	0	
Append3	UINT8	0	
Append4	UINT8	0	
Append5	UINT8	0	

Command 0x88

Data name	Data style	Value	Explain
CMD	UINT8	0x88	
Item_Add	UINT64		Device 64bits address
Data	UINT16		
SendType	UINT8	0x00	Using broadcast
		0x01	Sending to Item_Add

指令 0x88

Data name	Data style	Value	Explain
CMD	UINT8	0x89	
Item_Add	UINT64		Device 64bits address
Data	UINT16		
Parent_Add	UINT64		父節點的 64bits address
RSSI	**UINT8**		**RSSI when receiving command 0x88**

Samlpes

The 64 bit addess of Coordinator is 5152544300000073, the 64 bit addess of Device is 5152540000000710。

Changing the mode of Coordinator

0xCC 0xFF 0x12 0x28 0x51 0x52 0x54 0x43 0x00 0x00 0x00 0x73 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0xFF 0xCC

Changing the mode of Device

0xCC 0xFF 0x12 0x28 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x01 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0xFF 0xCC

Sending PING

0xCC 0xFF 0x0C 0x88 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x12 0x34 0x01 0xFF 0xCC

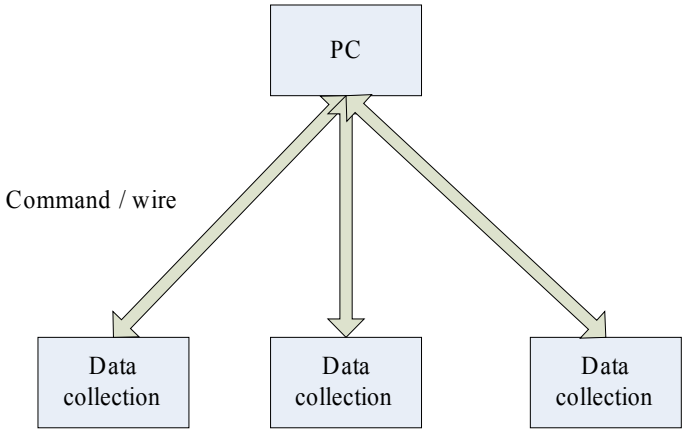
Receiving PING

0xCC 0xFF 0x14 0x89 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0x12 0x34 0x51 0x52 0x54 0x00 0x00 0x00 0x07 0x10 0xFF 0xFF 0xCC

6. Use case

Samlpe of user defined data

The user has one PC and sevaral data collections and connects PC with the data collections with the wire. The data colletion sends data to PC with a special format. This format includes the destination address. PC and data collections can process the command base on the destination address. Please see below:



Now user wants change the wire to Zigbee. There are tow ways to do that. User can choice one of them.

Blow is noun explain:

Host program: collecting data on the PC

DC program: the program running on the Data collection. It can collects thedata, send and receive data.

Control command: the potocol of communication between Host program and DC program

Destination address: the part of potocol. It can decide which one should prodess the command.

QR program: the new program whe user uses the Zigbee.

Method 1: User can change program

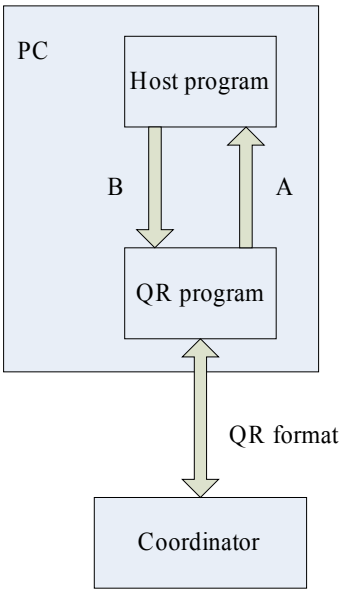
Scope

This samlpe fits that user can change the Host program and DC program to use the QR-format.

Parameters

The parameters are the default parameters on chap 5. User does't change them.

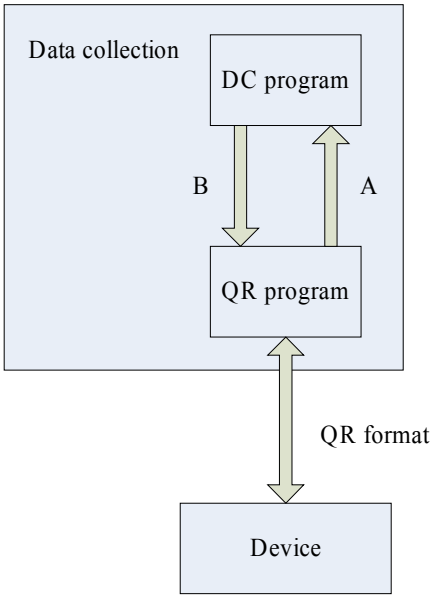
PC side



Above is the function block on PC side. "A" and "B" are control command. User should add the "QR program". It has three functions:

1. After building the network or finding a new device:
   a. Using command 0x77 to get the 64 bits address of all Devices.
   b. Building the mapping table between the 64bits address of Device and the destination address.
2. When the QR program receives the B:
   a. Cutting B into several pieces, each piece should be **small than 60 byters**.
   b. Packing these pieces into command 0x67 and sends to the Coordinator. QR program uses the destination address of cotrol command and the mapping table to find the DestMapAddress of command 0x67.
3. When the QR program receives command 0x67 from Coordinator
   a. Getting the payload form command 0x67
   b. Composing several payloads to full command A and sends to the Host porgram

Data collection side



Above is the function block on Data collection side. "A" and "B" are control command. User should add the "QR program". It has two functions:

1. When the QR program receives the B:
   a. Cutting B into several pieces, each piece should be **small than 60 byters**.
   b. Packing these pieces into command 0x67 and sends to the Coordinator.

QR program uses the destination address of cotrol command and the mapping table to find the DestMapAddress of command 0x67.

2. When the QR program receives command 0x67 from Coordinator
   a. Getting the payload form command 0x67
   b. Composing several payloads to full command A and sends to the Host porgram

Method 2: User can't chage the DC program

Scope

This samlpe fits that user can change the Host program but can't change the DC program.

Parameters

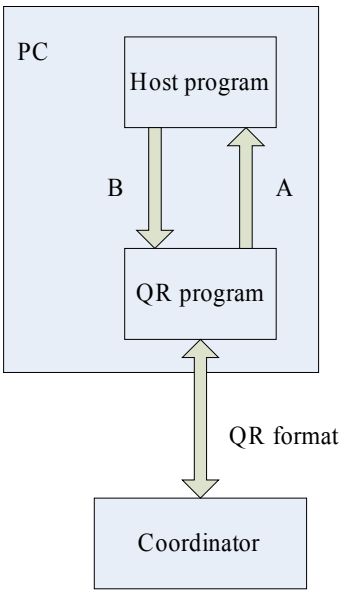
The parameters of Coordinator are the default parameters. User should use command 0x24 to change the type of UR on the Device side.

Command 0x24

Data name	Data style	Value	Explain
CMD	UINT8	0x24	
Item_Add	UINT64		Device address
Action	**UINT8**	**0x00**	**Update FLASH and Work immediately**
Trans	**UINT8**	**0x01**	**Using transparent mode**

BaudRate	UINT8	0x03	9600
Parity_Check	UINT8	0	None Check

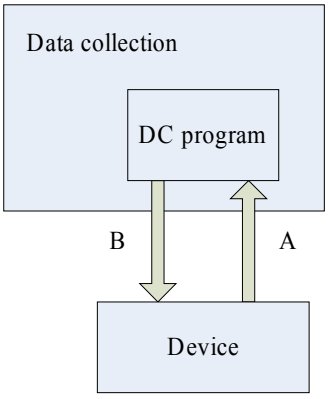
PC side



Above is the function block on PC side. "A" and "B" are control command. User should add the "QR program". It has three functions:

1. After building the network or finding a new device:
   a. Using command 0x77 to get the 64 bits address of all Devices.
   b. Building the mapping table between the 64bits address of Device and the destination address.
2. When the QR program receives the B:
   a. Cutting B into several pieces, each piece should be **small than 60 byters**.
   b. Packing these pieces into command 0x67 and sends to the Coordinator. QR program uses the destination address of cotrol command and the mapping table to find the DestMapAddress of command 0x67.
3. When the QR program receives command 0x67 from Coordinator
   a. Getting the payload form command 0x67
   b. Composing several payloads to full command A and sends to the Host porgram

Data collection side



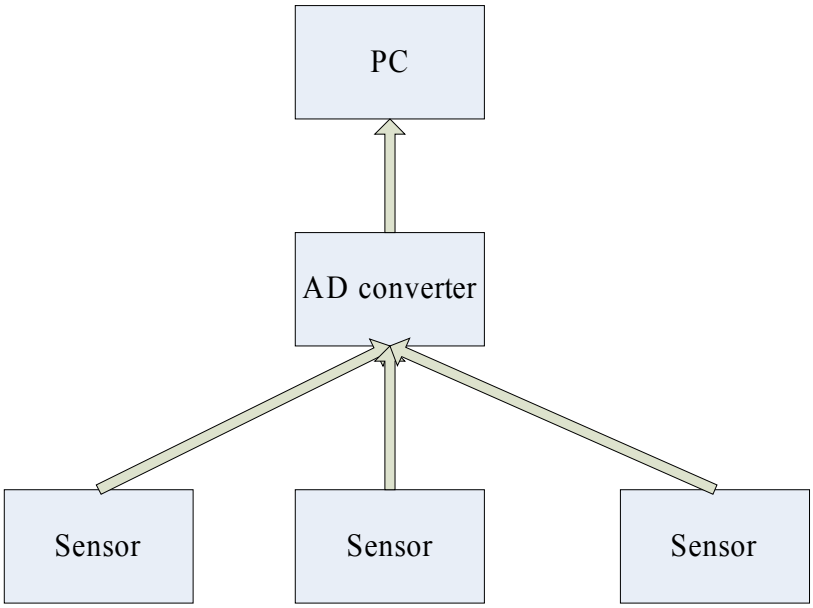
Above is the function block on PC side. "B" is control command. "A" is control command which be cutted several pieces.

1. When Device receives the B:
   a. Cutting the B, each pieces is small than **40** bytes.
   b. Packing the data into command 0x67 and sends to Coordinator.
2. When Device receive the command 0x67:
   a. Sending the payload of command 0x67 to the DC program.

Because Cooridnator cuts the control command and sends out, Device should have a wait time to wait the full command.

Sample of Sensor data

User has one PC, one AD converter and several sesnors. The PC, the AD converter and the sensor use the wire to connect to each other. An porgram which is running on the PC can read data for sensors. Show as below:



Now, user wants to change to wireless way

Blow is noun explain:

Host program: the program which is running on the PC reads the data of

sensors

QR program: the new program whe user uses the Zigbee.

Using zigbee module

Scope

This samlpe fits that user can change the Host program to use the QR-format.

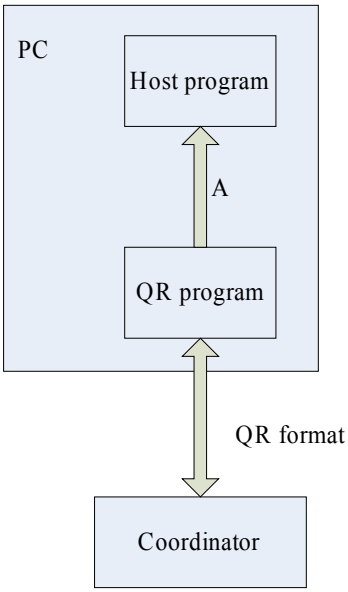
Parameters

The parameters of Coordinator are the default parameters. User should use command 0xB0 to change the type of Device.

Command 0xB0

Data name	Data style	Value	Explain
CMD	UINT8	0xB0	
Device_Add	UINT64		Device address
DeviceType	**UINT8**	**0x00**	**Sensor Device**
Data_Send_Time	UINT16	0x03E8	The time period when Device send data out actively. Uint : ms
SensorType	UINT8	0x00	CO Sensor
SensorPartNum	UINT8	0x00	Sensor type number

PC side

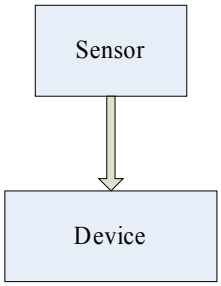


Above is the function block on PC side. "A" is sensor data. "QR program" is the new program which use should add. When QR program receives the command 0x62, it should do:

1. Getting the sensor data from command 0x62
2. Sending data to the Host program.

Sensor side

User will connect the sensor with the GPIO of Device.



The position of digital output, digital input, AD convert are as below:

Name	Position
Digital input	GPIO6
AD convert	GPIO7
Digital output	GPIO8

Smaple of sleep

You should know some thinngs in sleep mode. The things show as below:

1. Device can't receive RF single in sleep mode. So Device will wake regulaly.
2. The life time of the battery is negative with the frequency of regual wake of Device, the higer of frequency, the lower of battery life.
3. The life time of the battery is negative with the working time of Device, the longer of working time, the lower of battery life.
4. The response time is positive with the frequency of regual wake of Device, the higer of frequency, the faster of response.

User defined data

Using the session "Sample of user define data" as example, the first thing is that setting the parameters base on chapter 5. then sending data when Device wakes up.

PC side

When PC sends command to the data collection, it should do as:

1. Sending command 0x8A to Coordinator. The Item_Address is the 64bits address of Device which be connected with the data collection.

Command 0x8A

Data name	Data style	Value	Explain
CMD	UINT8	0x8A	
Coor_Add	UINT64		Coordinator 64bits address
WakeTime	**UINT8**		**Wake up time. Unit: second**
Item_Address	**UINT64**		**64bit address of the module which will be waked up**

2. Waiting the command 0x8B which comes from Coordinator. Check where the Item_Address in the Command 0x8B is the same as the Item_Address in

the Command 0x8A or not. It it is same, it means that the Device be waked.

Command 0x8B

Data name	Data style	Value	Explain
CMD	UINT8	0x8B	
ACK	UINT8	0x00	OK
Action	**UINT8**	**0x01**	**Module reports result**
Item_Address	**UINT64**		**64 bit address of Device which be waked up**

3. Sending data use no-sleep way. Notice: Device sleeps automatically; you should finish all work before it sleeps. The time is set at the WakeTime of Command 0x8A.

Data collection

When data colletion sends data automatically, it should use GPIO1 to wake Device up, then sends data using no-sleep way. Notice: Device sleeps automatically; the data collection should finish all work before it sleeps. The time is set at the EXTWakeTime of Command 0x05.

When data colletion sends data passively, it should spy GPIO2. If GPIO2 low to 0V, it means the Device wakes up. The data collection can send data using no-sleep way. Notice: Device sleeps automatically; the data collection should finish all work before it sleeps. The time is set at the Wakeup_Keep of Command 0x20.

Sensor data

When Sensor Device wakes up each time, it will read GPIO and send them toCoordinator.

Smaple of detecting modules

There are tow type of modules in detecting mode. One is Main-Point, the other is End-Poing.

Main-Point

The Main-Point includes one Coordinator or Device, one MCU and one display (like LCM).

The working process is:

1. The MCU sends the command 0x88 to the Main-point, the Main-point re-sends to the End-Point. The End-Poing sends the command 0x89 back. The MCU show the RSSI value of command 0x89 on LCM.

Parameters

To change the module into the detecting mode by command 0x28

Data name	Data style	Value	Explain
CMD	UINT8	0x28	
Item_Add	UINT64		Module 64bits address
ModuleType	**UINT8**	**0x01**	**For Device**

WorkMode	UINT8	0x00	Detecting mode
Low_Power	UINT8	0x00	0db
ConfirmMode	UINT8	0x01	
Append1	UINT8	0	
Append2	UINT8	0	
Append3	UINT8	0	
Append4	UINT8	0	
Append5	UINT8	0	

End-Point

The End-Point includes one Coordinator or Device

Parameters

Like Main-Point

Process of detecting

The process of detcting the modules is

1. Putting the End-Point in the place which already be setted.
2. Working to the place where you want to install the module and open the Main-Point. If the RSSI is bigger than the standard, it is a good place. If the RSSI is below the standard, you should find another place.

Repeat step 1 and 2 until all places are done.

Note of detecting

When you detect the places of the modules, you should open tow End-Points at least. So the Main-Point can see more than 2 End-Point. It means that your module has a back path to use. You can increase the stabilization of the network.

7. Appendix

Explanation of the parameter "Version" in command 0x13

The parameter "Version" is the main version of fimware. If the main version is changed, it means the format of command is changed. You can't mix two different man version of firmware together.

The parameter "RelaseVersion" is the change flag. We change it when we releae one version of firmawre. the meaning as below

Value	Explain
0xXXX0	QRZ-3000
0xXXX1	QRZ-2400
0xXXX2	QRZ-1000
0xXXX3	QRZ-1100-PA
0xXXX4	QRZ-3000-PA
0xXXX5	QRZ-3100

Release note

Release Version 0x05xx

1. Finish Router table method.
2. Finish Device.
3. Finish Coordinator inside confirm.
4. Finish 64 bits address as transform index.

Release Version 0x06xx

1. Coordiantor saves the 64 bits address of Device in FRESH ROM.
2. Finish child module of Coordinator.
3. Add nearby item command.

Release Version 0x07xx

1. Add setting parameters command.
2. Add Sensor Device.

Release Version 0x10xx

1. Formal Release。

Release Version 0x103x

1. The size of Payload in command RAWDATA magnifies to 60bytes.
2. Update the Router table renew rule.
3. UART of Device using Ring buffer。

Release Version 0x200x

1. The parameters of the Device can be changed using the network.
2. Changing the step of the Command 0x17 to 2 steps.
3. Changing the unit of checking network from minute to second in Command 0x1B.
4. Adding Command 0x01 for parameters of Zigbee.
5. The Bund of Coordiantor up to 115200bps。
6. User can change the bund of Deivce.
7. Adding sleep command.

Release Version 0x201x

1. Adding the encryption function.

Release Version 0x202x

1. Fix the bug about RX_FIFO in Coordinator.

Release Version 0x203x

1. Adding the serial number of Device data in Coordiantor. It can avoid to receive the repeat data.
2. Adding the functio that don't check the parent in Device.
3. Changing the time unit of sleep parameter form second to millisecond.
4. Adding the type of Firmware in Command.

Release Version 0x204x

1. Adding no-confirm mode
2. Adding detecting mode.
3. Adding RF power adjustment function
4. Fixing the Router table does't update when Device re-join the network
5. Fixing the power usage when Device wake up
6. Remove outside number
7. Adding the Fixed parent only function
8. User can change the time of wake up out side MCU
9. Fix the method of updating router table
10. Fix the command 0x88

QuadRep Electronics [T] Ltd.
16F-1, No. 75, Hsin Tai Wu Rd, Sec.1, His-Chih, Taipei, Taiwan
TEL: +886-2-26989933
FAX: +886-2-26989911
http:// www.quadrep.com.tw
http:// www.quadrep.com.cn