UART Interface Wlan Adapter Application Note

This document provides the SDK guideline for building up an application that use popular UART interface to connect with Ameba and send/receive data via Ameba WLAN interface.

Table of Contents

1 Introduction

This document introduces the implementation of uart adapter feature in Ameba. Smartphone and PC which has UART interface can transmit data to each other with Uart Adapter Feature in Ameba. The Setup process is quite simple that users just need to connect the PC and Ameba by UART and power on Ameba, after that users can use smartphone to transmit data and configure the Uart Parameter in Ameba.

2 Associate Uart Adapter to AP

The first time when power on, Uart Adapter will start simple configure process, use simple configure APP on smartphone to associate Uart Adapter to AP. The detail of how to use simple configure, please refer to <<AN0011 Realtek wlan simple configuration>> document.

After associate to AP successfully with simple configure, the AP information (ssid, channel, password, etc)will been written into Flash. Uart Adapter will read AP information from Flash and associate to the same AP automaticly when it power on again.

If the AP information has changed or Uart Adapter need to associate to another AP, push the GPIO reset button on Uart Adapter to erase the AP information in Flash and restart, Uart Adapter will then start simple configure process just like the first time it powers on.

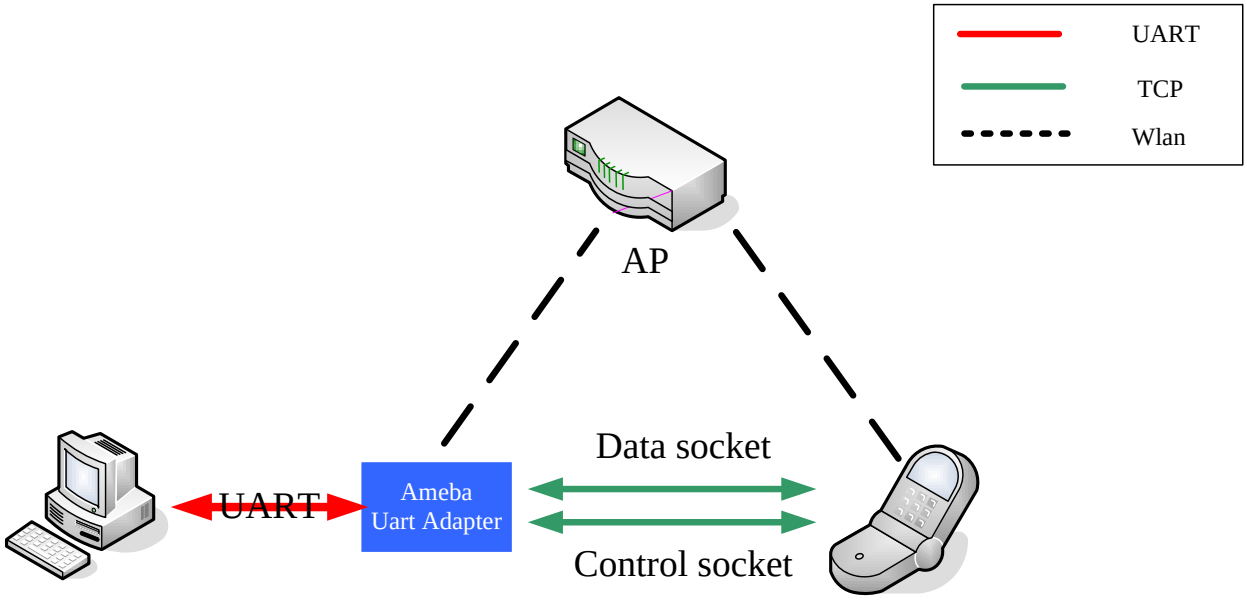
3 Uart Adapter Protocol

This sector will describe the protocol for communication between Uart Adapter and smartphone APP.

3.1.1 Architecture

There are two main function provided by Uart Adapter Feature. The first is chat data transmit between smartphone and PC or other device which has UART interface, the other is smartphone configure the parameters of UART on Uart Adapter, such as baudrate, stopbits, parity and so on.

3.1.1.1 Topology



PC or other device can communicate with Uart Adapter by UART.

Uart Adapter and smartphone associate with same AP.

There are two TCP connection between Uart Adapter and smartphone, one is data socket which will transmit chat data come from smartphone or PC; the other is control socket which will transmit the configuration information about UART parameters.

3.1.1.2 Discovery

MDNS is a protocol that resolves host names to IP addresses or Service name to IP address with layer4 type with port in local network that do not include a local name server. It is a zero configuration service, using essentially the same programming interfaces, packet formats and operating semantics as the unicast Domain Name System (DNS). the mDNS protocol is published as RFC 6762, uses IP multicast User Datagram Protocol (UDP) packets with 224.0.0.251 and port 5353.

After Uart Adapter associate to AP and get IP address in section 2, it will start 2 TCP server and start to listen, one is chat data TCP server with the ip address and port 5001, the other is control data TCP server with the ip address and port 6001.

At the same time, MDNS register process started. Uart Adapter will register 2 services in the local network with MDNS packets to advertise the chat data and control data TCP server information. The chat data service name is

ameba_[mac_adress]._uart_data._tcp.local with ip address and port 5001, the control data service name is ameba_[mac_adress]._uart_control._tcp.local with ip address and port 6001.

The smartphone which also has MDNS module can receive the MDNS packets and discover the service that Uart Adapter register. It can get the service name, service ip address and service port. With name it can identify the chat data service and the control data service.

With the information get from MDNS, smartphone can connect to the 2 TCP servers in Uart Adapter and then start to transmit chat data and control data with Uart Adapter.

3.1.1.3 Chat Data Process

3.1.1.3.1 TX to smartphone

The source of chat data is PC or Device, then chat data will been transmit from PC or device to Uart Adapter by UART. Uart Adapter receive the chat data in bytes and the chat data will be stored in buffer temporary. If there is no more chat data received in 50 microseconds, all the chat data received before will been send to the smartphone in TCP packets through chat data socket. If the chat data in buffer is more than 1400 bytes, it will also been send to the smartphone in TCP packets through chat data socket.

3.1.1.3.2 RX from smartphone

The chat data come from smartphone will been received by a thread which monitor the chat data and control data sockets by select function. The received chat data will then been sent to another thread by mailbox and been written to UART in that thread.

3.1.1.4 Control Data Process

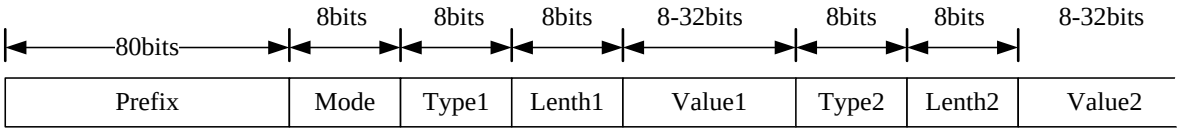
The control data socket is used to set and get the UART parameters in Uart Adapter by smartphone APP.

3.1.1.5 Format

Control data is wrapped in TCP packets and has specific formats. The most important is that the prefix of control data must been “AMEBA_UART”, otherwise the control data will been dropped even if it is received from control data socket.

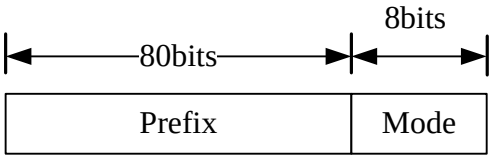
3.1.1.5.1 Set Request

From smartphone APP to Uart Adapter, this packet is used to set parameters of UART



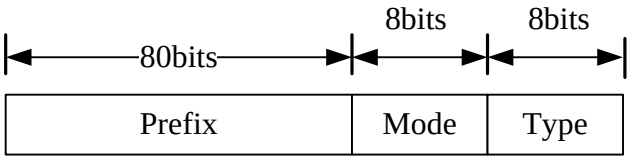
3.1.1.5.2 Set Reponse

From Uart Adapter to smartphone APP, indicate whether the set request before is success



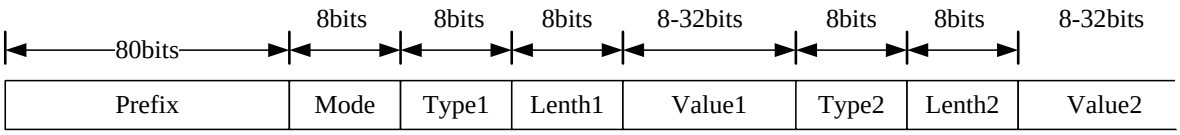
3.1.1.5.3 Get Request

From smartphone APP to Uart Adapter, this packet is used to get the UART information for Uart Adapter.



3.1.1.5.4 Get Reponse

From Uart Adapter to smartphone, this packet contain the UART information of Uart Adapter.



Note: in order to make the packets be parser earier by smartphone, every set response and get response packet will end up with “\n” in the end of packeten-UStp

3.1.1.5.4

Mode = 1: set response, send from Uart Adapter to smartphone

Mode = 2: get request, send from smartphone to Uart Adapter

Mode = 3: get response, send from Uart Adapter to smartphone

PS: the range 0~3 is just the Hex number in the field, for example: Mode = 2 will fill the field with 0x02, Mode = 3 will fill the field with 0x03

Type: indicate the type of parameters, length is 1 byte and use bitmap which means every bit in the byte will indicate one kind of parameter type.

Bit 0 = 1: UART BaudRate (9600, 38400, 115200…)
Bit 1 = 1: UART WordLen (only support 7 and 8)
Bit 2 = 1: UART Parity (0-none, 1-odd, 2-even)
Bit 3 = 1: UART StopBit (0-1stop bit , 1-2 stop bit)
Bit 4 – Bit 7: not support yet, for later extended

There are two different situations for **Type**, one is **set request** which started by smartphone, if the request want to set UART Baudrate, fill the **Type** with 0x01, if the request want to set UART Parity, fill the **Type** with 0x04, if the request want to set more than one parameters one time, use the **TLV** format to fill the packet in sequence, each **TLV** represent one parameter and its value. **Get response** started by Uart Adapter has the same format with **set request**.

The other situation is in **get request** which also started by smartphone. No **TLV** is needed and there is only one **Type** in get request packet. If get request want to get UART Baudrate, fill the **Type** with 0x01, if get request want to get UART Baudrate and Parity, fill the **Type** with 0x05 which means bit 0 and bit 2 is set. If get request want to get all the parameters supported, fill the **Type** with 0x0F. So there is only one **Type** field in the get request packets.

Lenth: indicate the length of parameters with bytes in units. The length of **Lenth** is fixed 1 byte. The range of Lenth is 1-4:

Lenth = 0x01: the value length is 1 byte.

Lenth = 0x02: the value length is 2 byte.

Lenth = 0x03: the value length is 3 byte.

Lenth = 0x04: the value length is 4 byte.

Value: the value of parameters, the length is depend on **Lenth.**

Lenth = 0x01: value range [0-0xFF].

Lenth = 0x02: value range [0-0xFFFF].

Lenth = 0x03: value range [0-0xFFFFFF].

Lenth = 0x04: value range [0-0xFFFFFFFF].

There are two kind of **Value**, one is in **set request** or **get response**, **Value** means the value of UART parameters, the other is in **set response**, **Value** means set result, **Value** = 0 means set successful, otherwise means set error.

	usage	type	range
Prefix	Identify the control data packets	String	ASICII of "AMEBA_UART"
Mode	Option type	Uint8	0x00-0x03
Type	Parameter type	Uint8	0x00-0x1F
Lenth	Parameter length	Uint8	0x01-0x04
Value	Parameter value	Decide by Lenth	0x0-0xFFFFFFFF

3.1.1.6 Illustration

3.1.1.7 Instance 1

Smartphone want to set the Baudrate of Uart Adapter to 115200

1, smartphone sent set request control data as follow through control data socket, if no set response returned, smartphone can send set request again.

| AMEBA_UART | 0 | 1 | 4 | 115200 |

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	set	request
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54		0x00

baudrate	lenth 4byte	Value 115200
0x01	0x04	0x00 0xC2 0x01 0x00（fill in reverse for byte-order）

2, Uart Adapter receive the set request. It will set the Baudrate and send back set response packet as follow. Smartphone receive the set response packet and consider the set option is successful.(error code is not supported now)

| AMEBA_UART | 1 | \n

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	set response
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x01

3.1.1.8 Instance 2

Smartphone want to get the UART Baudrate information of Uart Adapter.

1, Smartphone send get request control data as follow through control data socket, if no get response returned, smartphone can send get request again.

| AMEBA_UART | 2 | 1 |

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	get request
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x02

baudrate
0x01

2, Uart Adapter receive the get request and send back get response as follow, smartphone will receive the response and get the baudrate information

---

| AMEBA_UART | 3 | 1 | 4 | 115200 |\n

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	get response
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x03

baudrate	Lenth 4byte	Value 115200
0x01	0x04	0x00 0xC2 0x01 0x00

3.1.1.9 Instance 3

Smartphone want to set the Baudrate to 115200 and WordLen to 7

1, smartphone sent set request control data as follow through control data socket, if no set response returned, smartphone can send set request again.

| AMEBA_UART | 0 | 1 | 4 | 9600 | 2 | 1 | 7|

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	set	request
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x00	

baudrate	Lenth 4byte	Value 9600	wordlen
0x01	0x04	0x80 0x25 0x00 0x00	0x02

Lenth 1byte	Value 7
0x01	0x07

2, Uart Adapter receive the set request. It will set the Baudrate and WordLen and send back set response packet as follow. Smartphone receive the set response packet and consider the set option is successful.(error code is not supported now)

| AMEBA_UART | 1 | \n

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	set response
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x01

3.1.1.10 Instance 4

Smartphone want to get the UART Baudrate and Stopbits information of Uart Adapter.

1, Smartphone send get request control data as follow through control data socket, if no get response returned, smartphone can send get request again.

| AMEBA_UART | 2 | 9 |

	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	get request
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x02

baudrate & stopbit
0x09

2, Uart Adapter receive the get request and send back get response as follow, smartphone will receive the response and get the Baudrate and Stopbits informations

---

| AMEBA_UART | 3 | 1 | 4 | 115200 | 8 | 1 | 2 |\n

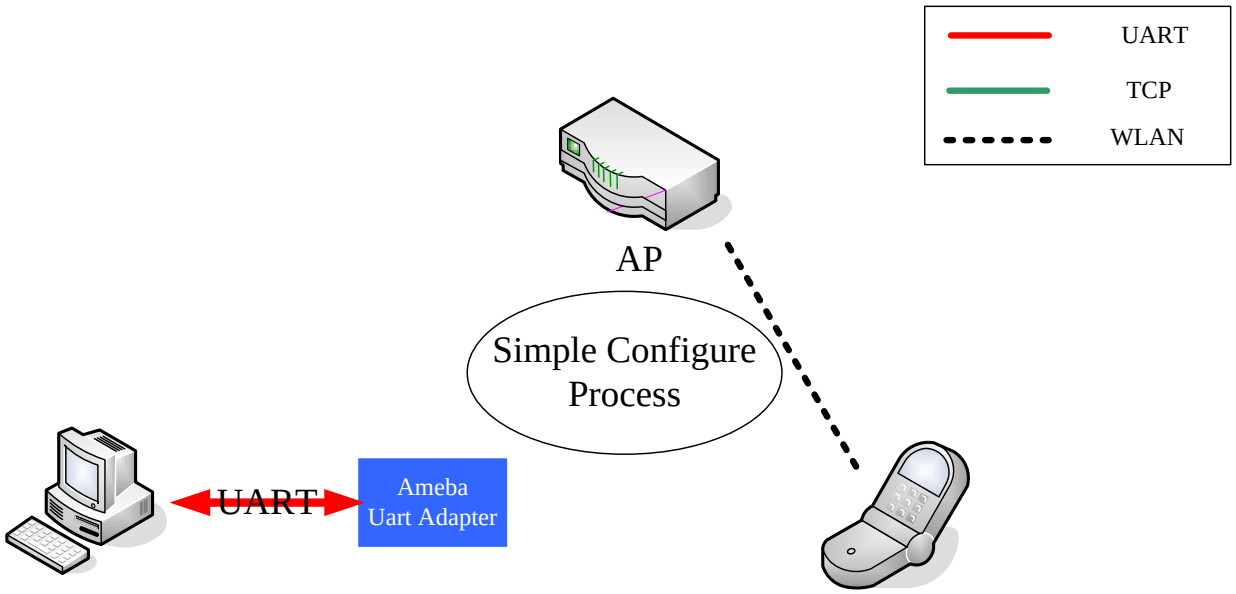
	"A"	"M"	"E"	"B"	"A"	"_"	"U"	"A"	"R"	"T"	get response
Actual Data	0x41	0x4D	0x45	0x42	0x41	0x5F	0x55	0x41	0x52	0x54	0x03

baudrate	Lenth 4byte	Value 115200	stopbit
0x01	0x04	0x00 0xC2 0x01 0x00	0x08

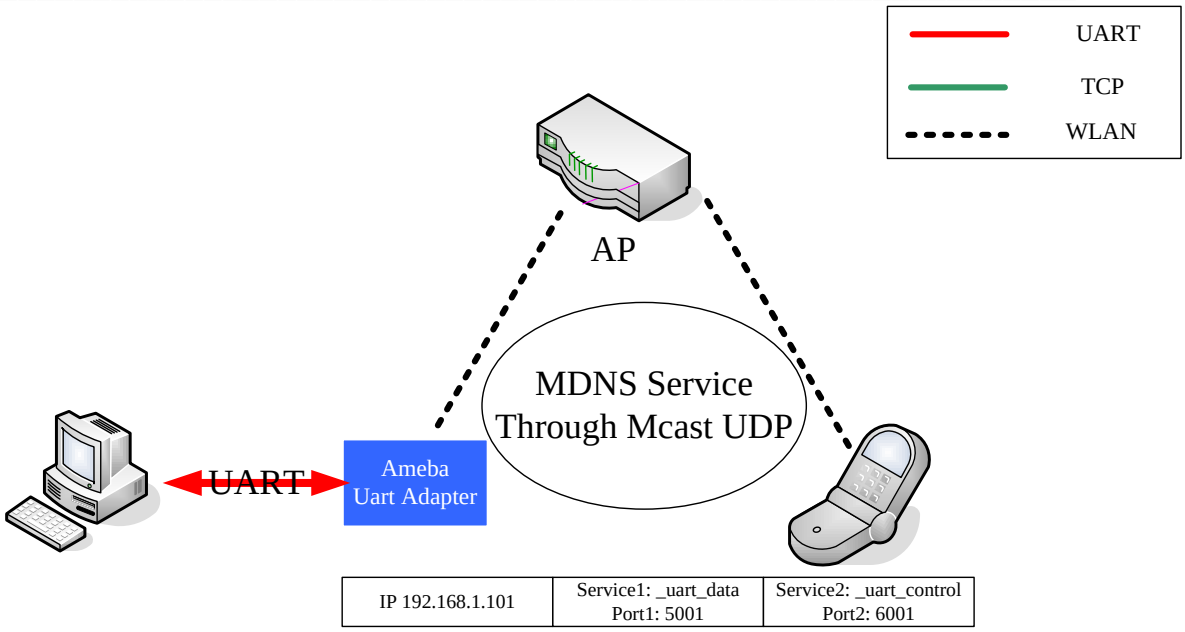
Lenth 1byte	Value 2
0x01	0x02

3.1.2 Flow Chart

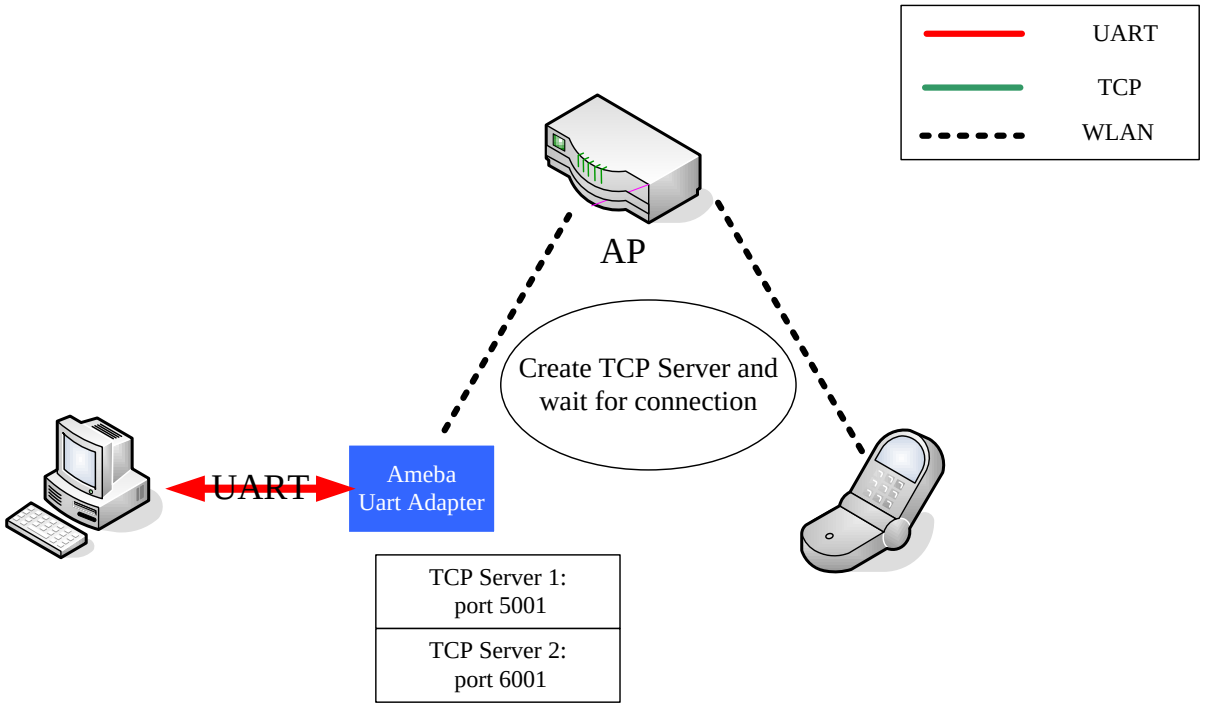
Step1: Uart Adapter enter simple configure process when power on and wait to associate to AP by smartphone



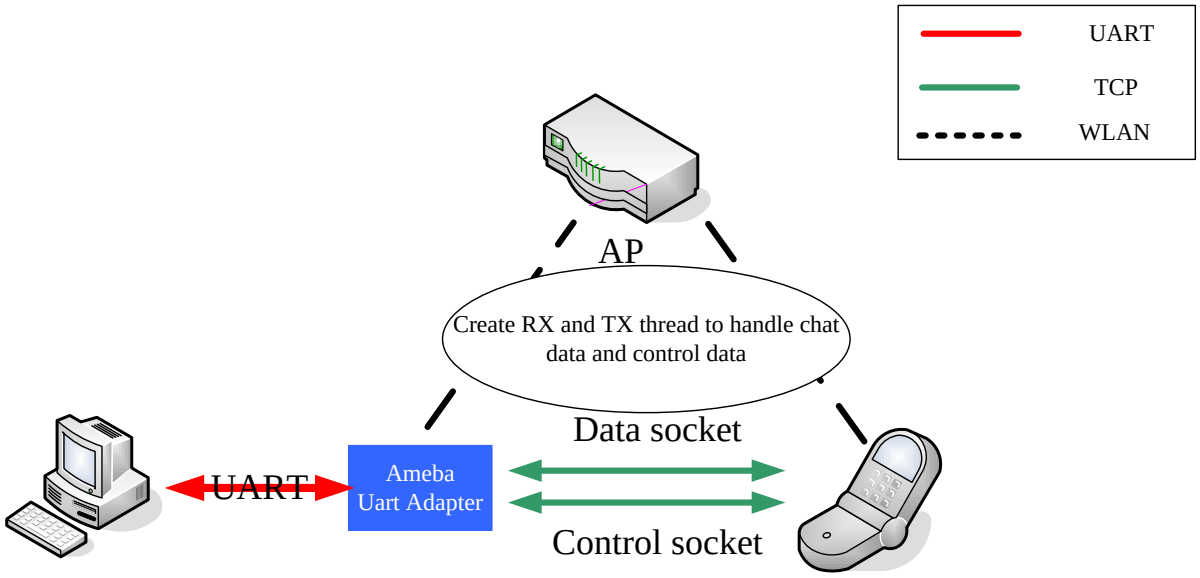
Step2: Uart Adapter advertise its chat data and control data TCP server information by MDNS.



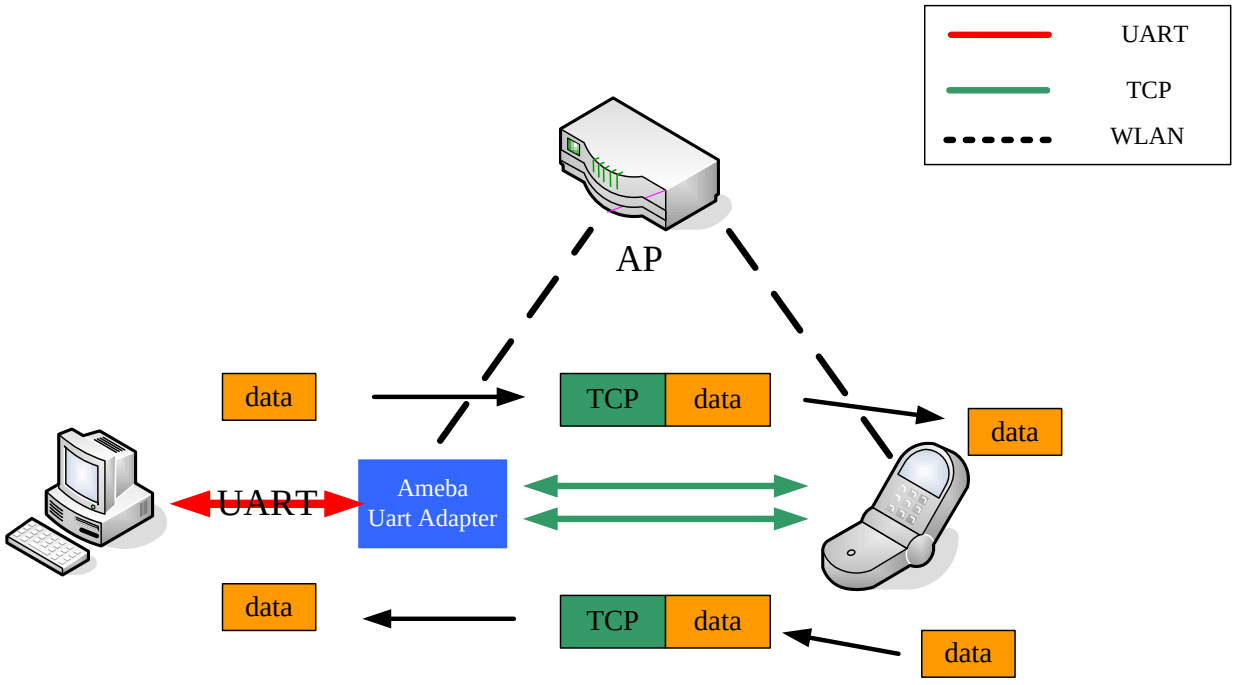
Step3: Uart Adapter start two TCP server and wait for connection come from smartphone



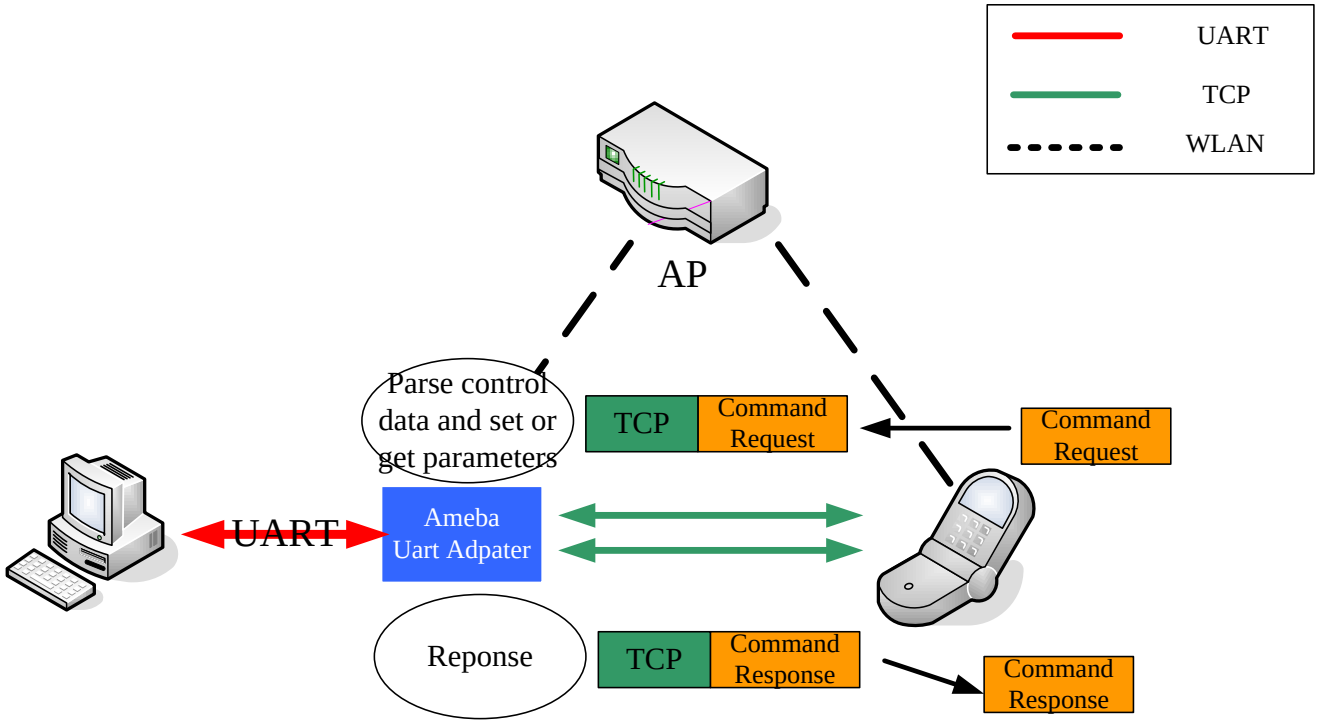
Step4: Create RX and TX thread after connection has been established.



Step5: Chat data transmit



Step6: Control data transmit



3.1.3 Setup guide

3.1.3.1 Complie guide

To enable Uart Adapter in Ameba, please make sure the macro as follow are configured correctly. Please configure CONFIG_EXAMPLE_UART_ADAPTER to 1 in platform_opt.h. Please also configure LWIP_UART_ADAPTER to 1 in lwipopt.h

```
/* platform_opts.h *.
#define CONFIG_EXAMPLE_UART_ADAPTER          1
#if CONFIG_EXAMPLE_UART_ADAPTER
#undef CONFIG_EXAMPLE_WLAN_FAST_CONNECT
#define CONFIG_EXAMPLE_WLAN_FAST_CONNECT 1
#undef CONFIG_EXAMPLE_MDNS
#define CONFIG_EXAMPLE_MDNS    1
#endif

/* lwipopts.h */
#define LWIP_UART_ADAPTER              1

#if LWIP_UART_ADAPTER
#undef  LWIP_IGMP
```

```
#define LWIP_IGMP                       1

#undef  LWIP_SO_SNDTIMEO
#define LWIP_SO_SNDTIMEO                1

#undef  SO_REUSE
#define SO_REUSE                        1

#undef  LWIP_TCP_KEEPALIVE
#define LWIP_TCP_KEEPALIVE                      1

#define MEMP_NUM_NETCONN            10

#define TCP_KEEPIDLE_DEFAULT                10000UL
#define TCP_KEEPINTVL_DEFAULT               1000UL
#define TCP_KEEPCNT_DEFAULT                 10U
#endif
```

Once startup correctly, the output of log uart should be as follows:

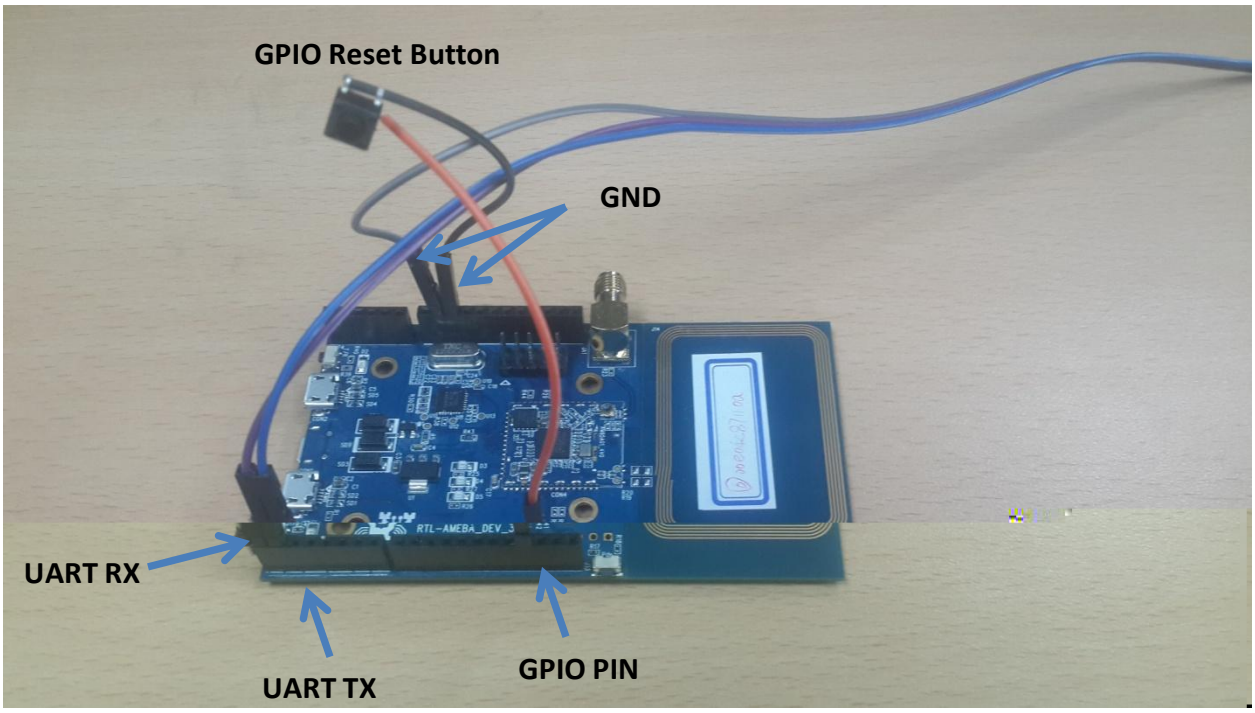
```
===== Enter Image 2 ====
#
Initializing WIFI ...
Start LOG SERVICE MODE

#
==============>uartadapter_init()
RTL8195A[HAL]: ISR 8 had been allocated!!!
RTL8195A[HAL]: ISR 8 had been allocated!!!

WIFI initialized

init_thread(47), Available heap 0x5b10
No AP Profile read from FLASH, start simple configure

Switch to channel(2)

Switch to channel(3)

Switch to channel(4)

Switch to channel(5)

Switch to channel(6)

Switch to channel(7)

Switch to channel(8)

Switch to channel(9)

Switch to channel(10)
```

3.1.3.2 Peripheral guide

There are two Peripheral need to setup: UART0 and GPIO, for detail usage please refer to <<UM0035 Realtek ameba1 peripheral vertification>> document.

UART0: use to receive and send data through uart.

GPIO: use to erase the AP information in Flash and restart.



3.1.3.3 Power Save

Uart adapter can support power save mode and wakeup by Uart and TCP. To enable power save mode please set UA_PS_ENABLE to 1 in uart_adpater.h.

There are two ways to wakeup system by UART

(A) Select a UART RX pin which is also an a GPIO interrupt pin
(B) Parallel UART RX with another GPIO interrupt pin.

The UA_GPIO_WAKEUP_PIN is used to wakeup system by Uart RX event in solution B, users can use any GPIO interrupt PIN as UA_GPIO_WAKEUP_PIN, the user guide and more detail of power save mode please refer to << AN0045 Realtek Ameba-1 power modes.docx>>

```
/* uart_adapter.h */
```

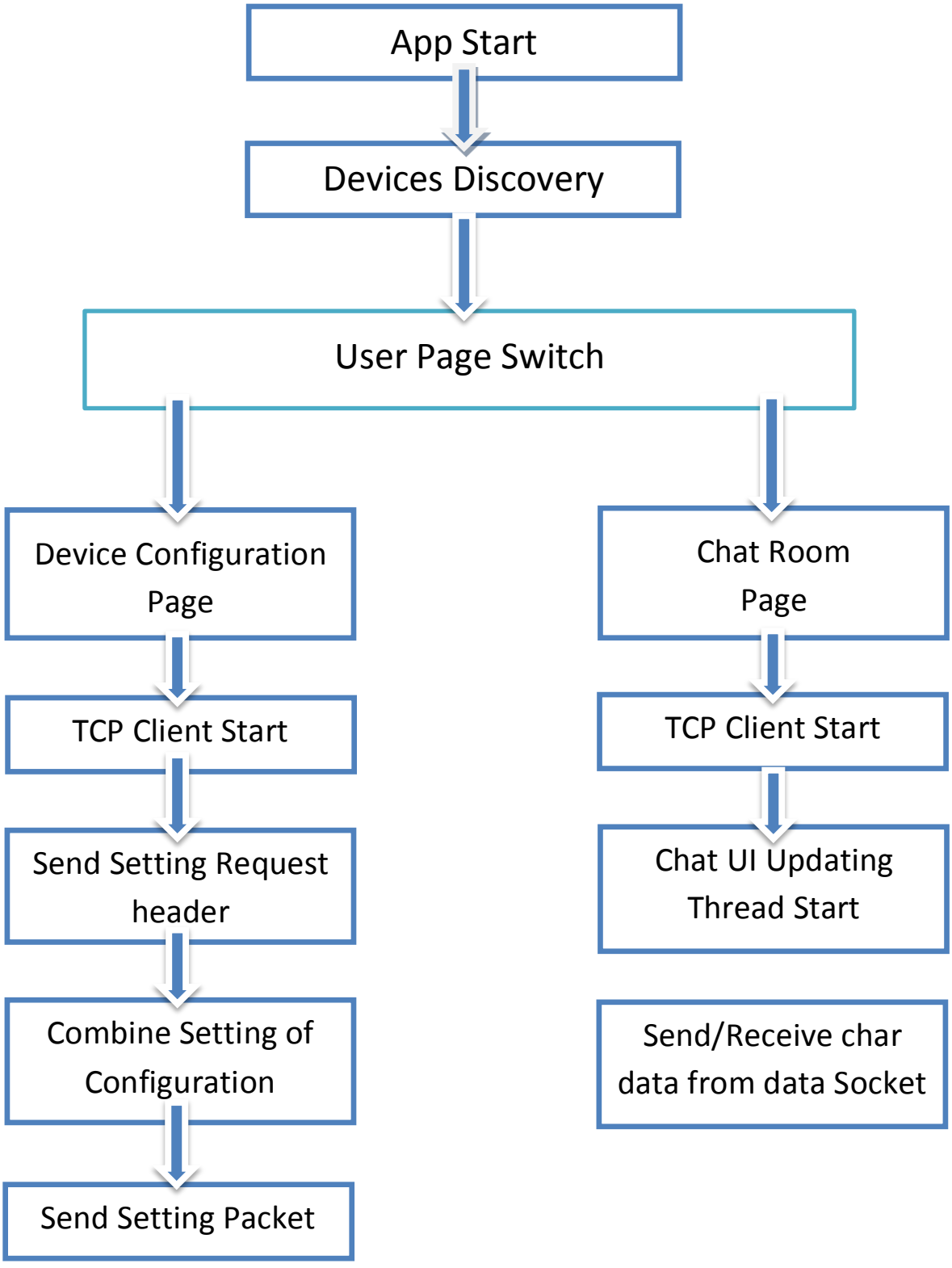
```
#define     UA_UART_TX_PIN      PA_7
#define     UA_UART_RX_PIN      PA_6

#define     UA_GPIO_LED_PIN         PC_5
#define     UA_GPIO_IRQ_PIN         PC_4

#define     UA_CONTROL_PREFIX   "AMEBA_UART"

#define     UA_PS_ENABLE        1
#define     UA_GPIO_WAKEUP_PIN      PC_3
#define     UA_WAKELOCK         WAKELOCK_USER_BASE
```

4 APP

The smartphone application implemented three main functions provided by UART Adapter Feature. The first is discovering the Ameba devices at a local network environment. The second is configuring the parameters of UART on UART Adapter. The finally function is chat data transmit between smartphone and PC through Ameba device.

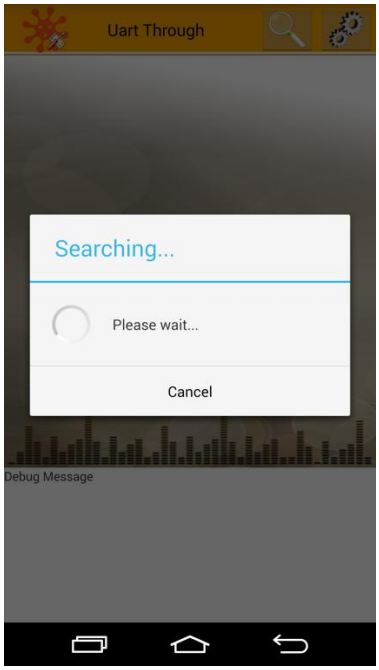
4.1 Android

4.1.1 Architecture

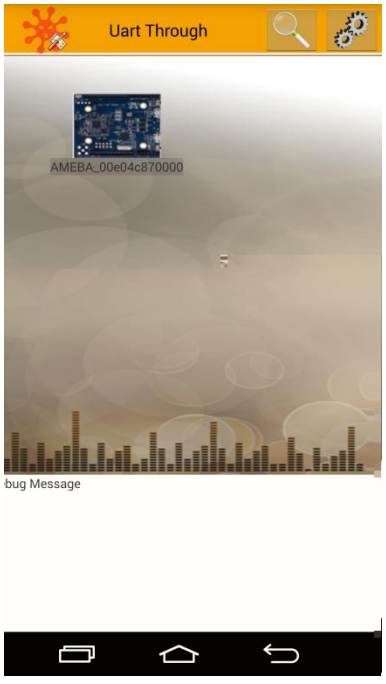


4.1.2 Setup guide

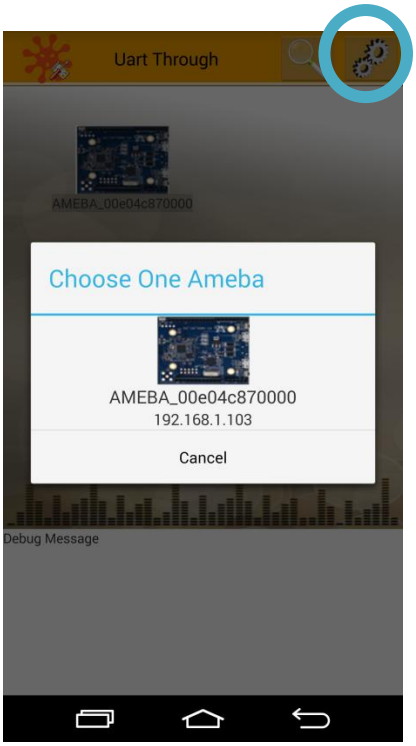
Step 1. Run UartThrough App. The device will auto-discovery when App is first run.



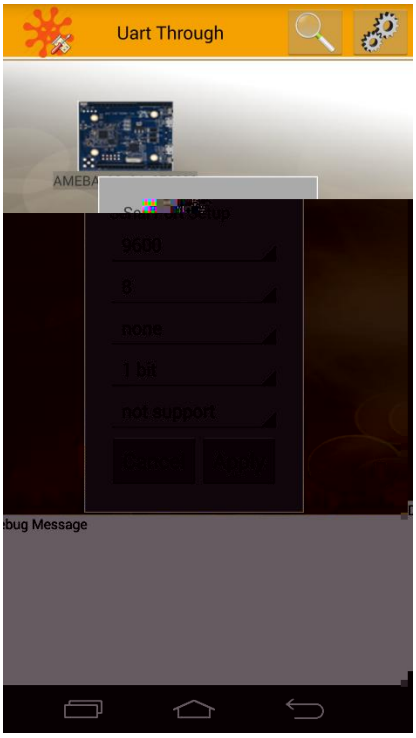
Step 2. The Ameba device will be found if the device was connected in a local network.



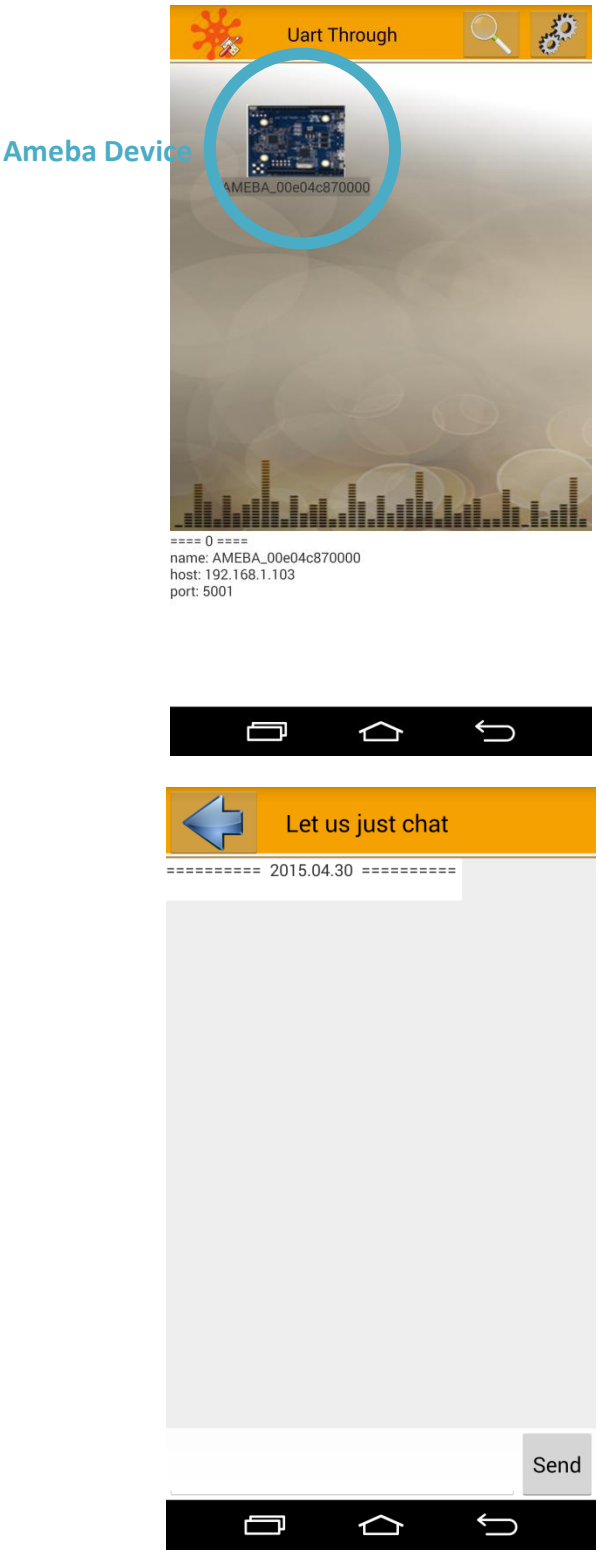
Step 3. Press the setting button on the upper right corner. The found devices list will pop up on Screen. Touch one device than the setting page will be show.



Step 4. Set the UART parameters of Ameba device.

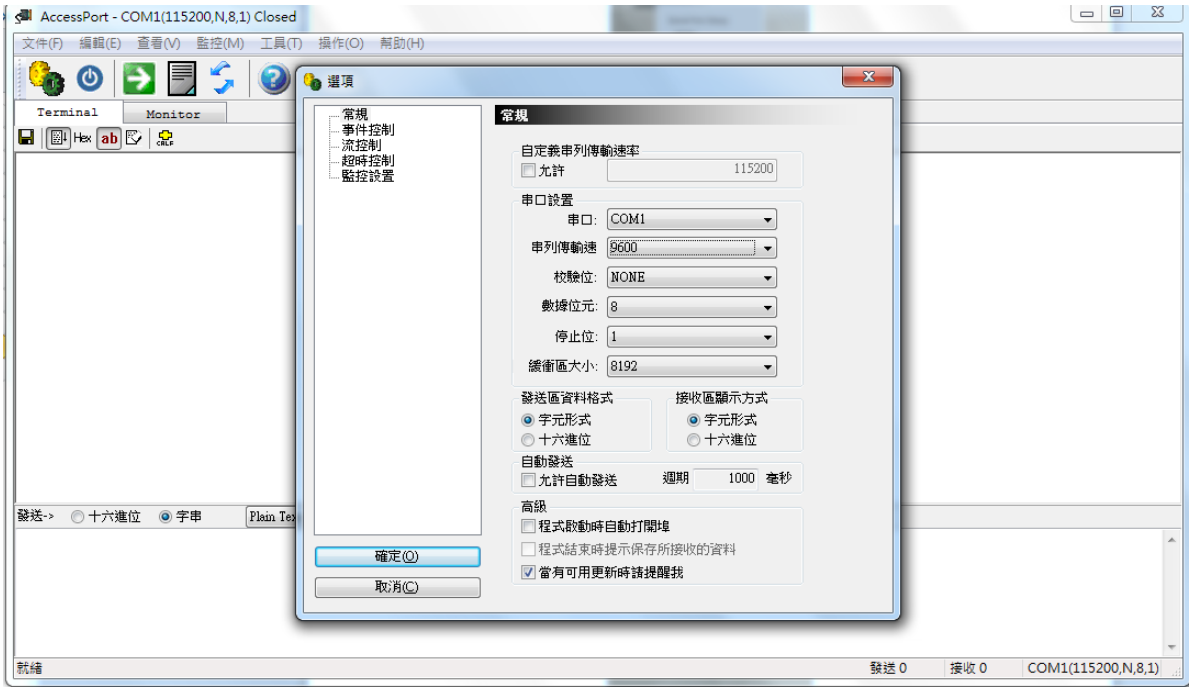


Step 5. Return the main page and press one Ameba device. Than next chat page will be show.

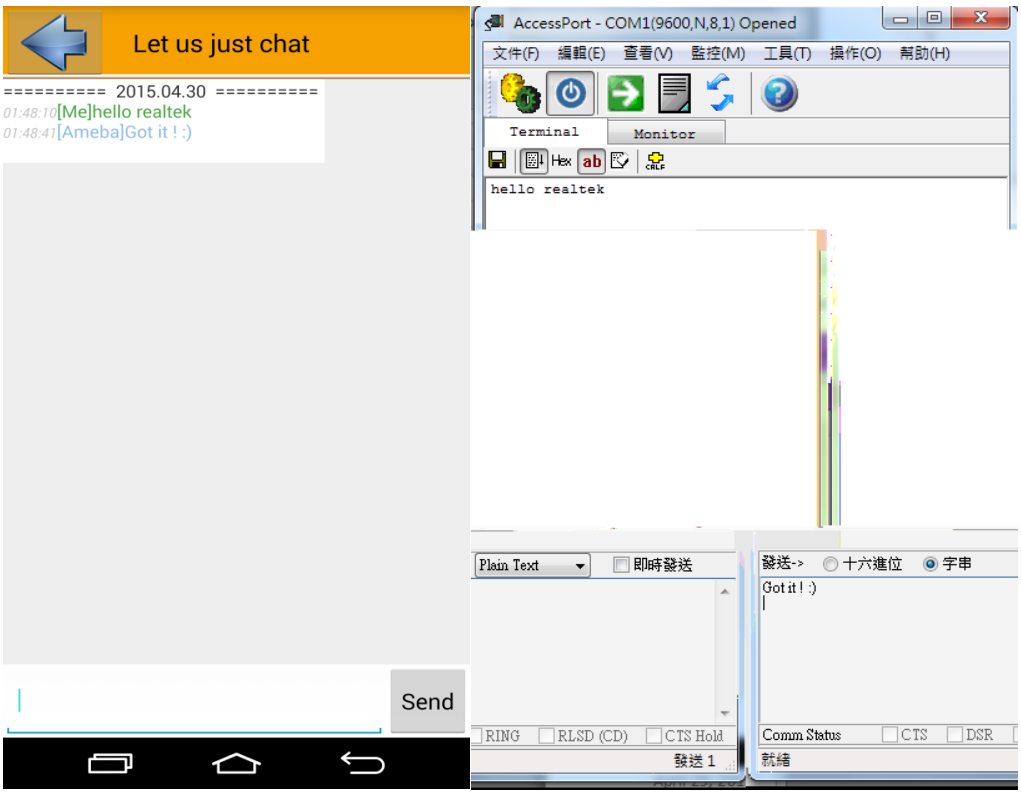


Step 6. Run an UART tool on computer. Ex: AccessPort

The UART setting of AccessPort tool must be same as the setting changed on App.



Step 7. Now the chat function can be running.



4.2 iOS

To be continued