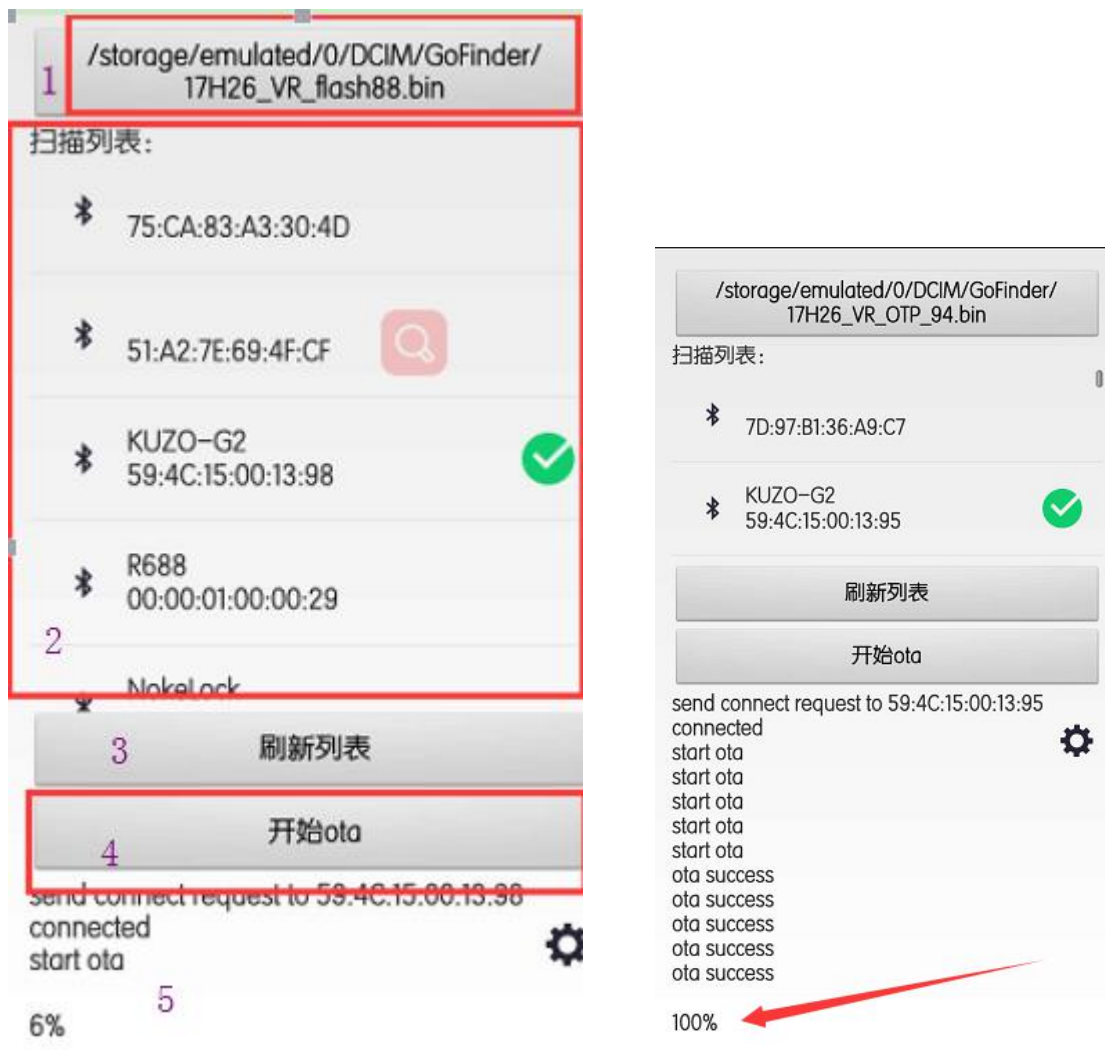


A brief introduction of update program Over The Air

Step 1: install this application



Step 2: open this app in your mobile phone:



In this figure ,you can see 5 modules each with a number on it :

Number 1 : target "*.bin" file

Description: choose the path contains the target "*.bin" file you want to update .

Number 2 : the blue tooth device list

Description: you can choose the idea device from this list which all the blue tooth device in advertising state are listed in this list .

Description: you can update the blue tooth device in advertising state after click this button.

Description: click this button will trigger OTA(Over the Air) transferring event.

Description: this part shows the procedure and rate of process during OTA event .

1 . to use the OTA attribute ,you have to enable it in the attribute list below ,and you have to make sure you are compiling and loading the `**_flash.bin` into the chip .

```

/*****
OTA (Services)
*****/
#ifdef OTA_ENABLE
    {4,2,16, (u8*)(&my_primaryServiceUUID), (u8*)(&ota_service_uuid)},
    {0,2,1, (u8*)(&my_characterUUID), (u8*)(&PROP_READ_WRITE_NORSP_NOTIFY)}, //prop
    {0,16,20, (u8*)(&ota_write_char_uuid), ota_data}, //value
    {0,2,2, (u8*)(&clientCharacterCfgUUID), (u8*)(&generalValInCCC)}, //value
#endif
};

```

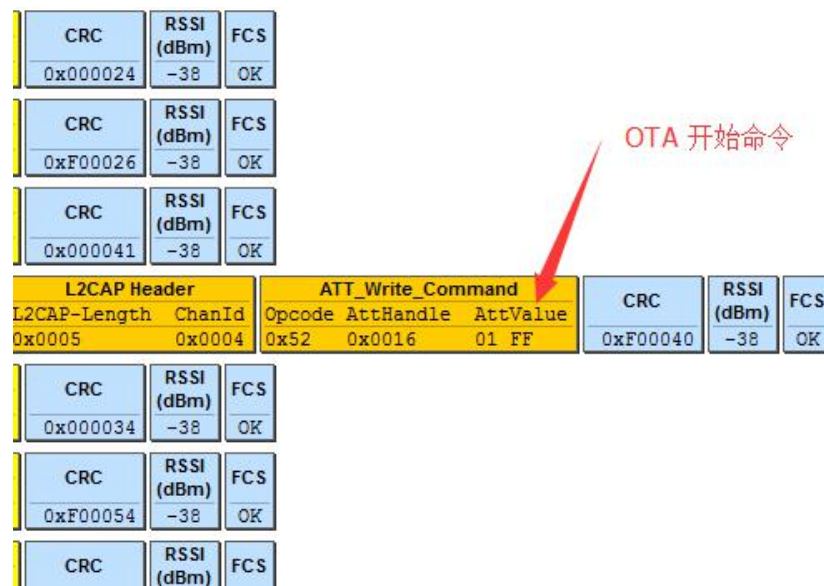
Besides, you must add the following code to `att_write_cb()` so that the chip will be able to process the data send through OTA.

```

812     }
813     else if((write_value == 1) || (write_value == 2)){
814         buzzer_enter_mode(2);
815         led_enter_mode(2);
816         alert_time =1;
817     }
818 }
819 }
820 // if(MY_BATVAL == att_handle){
821 //     proc_battery_value();
822 // }
823
824 #if(OTA_ENABLE)
825     if(att_req->handle == 44){
826         u8 result=otaWrite(att_req);
827         if(result){
828
829         }
830         return ATT_HANDLED;
831     }
832 #endif
833     ret_handle = att_handle;
834
835 // return ATT_NO_HANDLED;////*****
836 return ATT_HANDLED;

```

2.The OTA app will start the OTA process by sending " 0xff01" to the device which support the OTA service.



3.APP send target bin file through the ATT_OP_WRITE_CMD (0x52) , the data structure is list below:

DataArray [23]	Description
0~1	SerialNumber(start from 0x0000)
2~17	16byte data of new bin file
18~19	CRC Value of previous 18 bytes
20~22	Reserved

4. And the packets during OTA transferring are captured as below :

Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00028	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	00 00	0E 80 01 03 00 00 00 00 00 4B 4E 4C 54 20 01 88 00 98 A5															
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000029	-38	OK																
Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00027	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	01 00	76 80 00 00 00 00 00 00 00 84 3B 00 00 00 00 00 85 6E															
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000047	-38	OK																
Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00026	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	02 00	31 08 32 09 32 0A 91 02 02 CA 08 50 04 B1 FA 87 8C 26															
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000037	-38	OK																
Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00031	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	03 00	20 08 C0 6B 21 08 85 06 1F 08 C0 6B 20 08 85 06 35 04															
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000037	-38	OK																
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0xF00039	-38	OK																
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000034	-38	OK																
Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00026	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	04 00	00 A0 20 09 20 0A 91 02 02 CA 08 50 04 B1 FA 87 8B 7E															
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000039	-38	OK																
Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00034	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	05 00	1F 09 20 0A 91 02 02 CA 08 50 04 B1 FA 87 1B 09 05 52															
Header	CRC	RSSI (dBm)	FCS																			
0	MD PDU-Length	0	0	0x000038	-38	OK																
Header	L2CAP Header			ATT_Write_Command																CRC	RSSI (dBm)	FCS
0	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue											0xF00028	-38	OK			
0	27	0x0017	0x0004	0x52	0x0016	06 00	1D 08 08 40 01 B0 48 40 3F 23 18 F3 18 58 25 2B D4 BF															

5. The OTA app will stop the OTA process by sending " 0xFF02" to the device.

gth	CRC	RSSI (dBm)	FCS					
	0x000030	-38	OK					
ch	L2CAP Header		ATT_Write_Command			CRC	RSSI (dBm)	FCS
	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue	0xF00029	-38	OK
	0x0005	0x0004	0x52	0x0016	02 FF			
gth	CRC	RSSI (dBm)	FCS					
	0x000038	-38	OK					
gth	CRC	RSSI (dBm)	FCS					
	0xF00023	-38	OK					
	CRC	RSSI (dBm)	FCS					

OTA 升级结束命令

OTA 升级结束命令

1. After Slave receives the OTA_END command, slave Run Ota_boot.bin to reboot, MCU will flash 0x00000 address of the Old_firmware.bin in the previous part of the command moved to SRAM from the beginning of the 0x808000, run old_ Firmware.bin CSTARTUP.S. The corresponding boot code, which detects the value of the Boot_flag on the Flash 0x73000, and discovers that the value is 0xA5, which is no longer running the corresponding normal old_ Firmware.bin code, but will move 0x72000~0x72600 area 1.5K Ota_boot.bin from flash to the SRAM 0x808000~0x808600 place, after removal, reset MCU (reset Just let the MCU start from the SRAM 0x808000 address, it

will not move code from flash to SRAM. At this time, the MCU from 0x808000 to start again, the equivalent of running the ota_boot.bin function.

7. ota_boot update code, reboot ota_boot.bin after running, from the beginning of the Flash 0x20000 read new_firmware.bin content, and write to the beginning of the flash 0x00000 corresponding address, the equivalent of New_Firmware.bin is fully updated to the 0 address of Flash . After the update is completed, set the value of the Boot_flag on the Flash 0x73000 to 0x00, and reboot MCU.

8) New_firmware.bin normal operation MCU will reboot once again , from the Flash 0 beginning address move to the of the SRAM 0x808000, and detect the Boot_flag value is not 0xa5, start the normal slave function, the new_firmware.bin similar to the previous old_Firmware.bin, also has OTA function, can start OTA mode update code again (the latest code to be downloaded to the Ota_master Flash 0x20000 address).