Upgrade Specification for FT6336GU

Upgrade Specification for FT6336GU	
Project name	Touch panel
Version	1.0
Release date	Jul 3, 2015
Owner	ZhangXiaoyi
Classification	Confidential
Approval	

3/F, Kingdom Sci-Tech Building, 5th South Gaoxin Avenue, Hi-tech Park,
Nanshan District, Shenzhen, Guangdong, China
ZIP: 518057
T +86 755 26588222
F +86 755 26712499
E support@focaltech-systems.com

www.focaltech-systems.com

Revision History

Date	Version	List of changes	Author + Signature
2014-12-10	0.1	Initial draft	Js Liao
2015-07-03	1.0	Chip named FT6336GU	ZhangXiaoyi

Table of Contents

1 Introduction

This document descripts the upgrade flow of FT6416/FT6426 IC series. The whole program of FT6416/FT6426 is divided into two parts, one is bootloader part and the other is application part. And both parts will be burned into flash when we run the download operation. When FT6416/FT6426 boot up, the bootloader part will run at first, then there will be a timer with 30ms time out started. The upgrade flow will be triggered if the bootloader received specific upgrade command within this timer, otherwise it will switch to application flow part.

2 Communication

The upgrade flow communication based on the standard IIC protocol. And the default IIC slave address is 0x70. The IIC slave address in bootloader and in application should be the same one. To ensure this, bootloader and application will both get the IIC slave address from a specific flash address (0x07B0).

3 Basic Flow

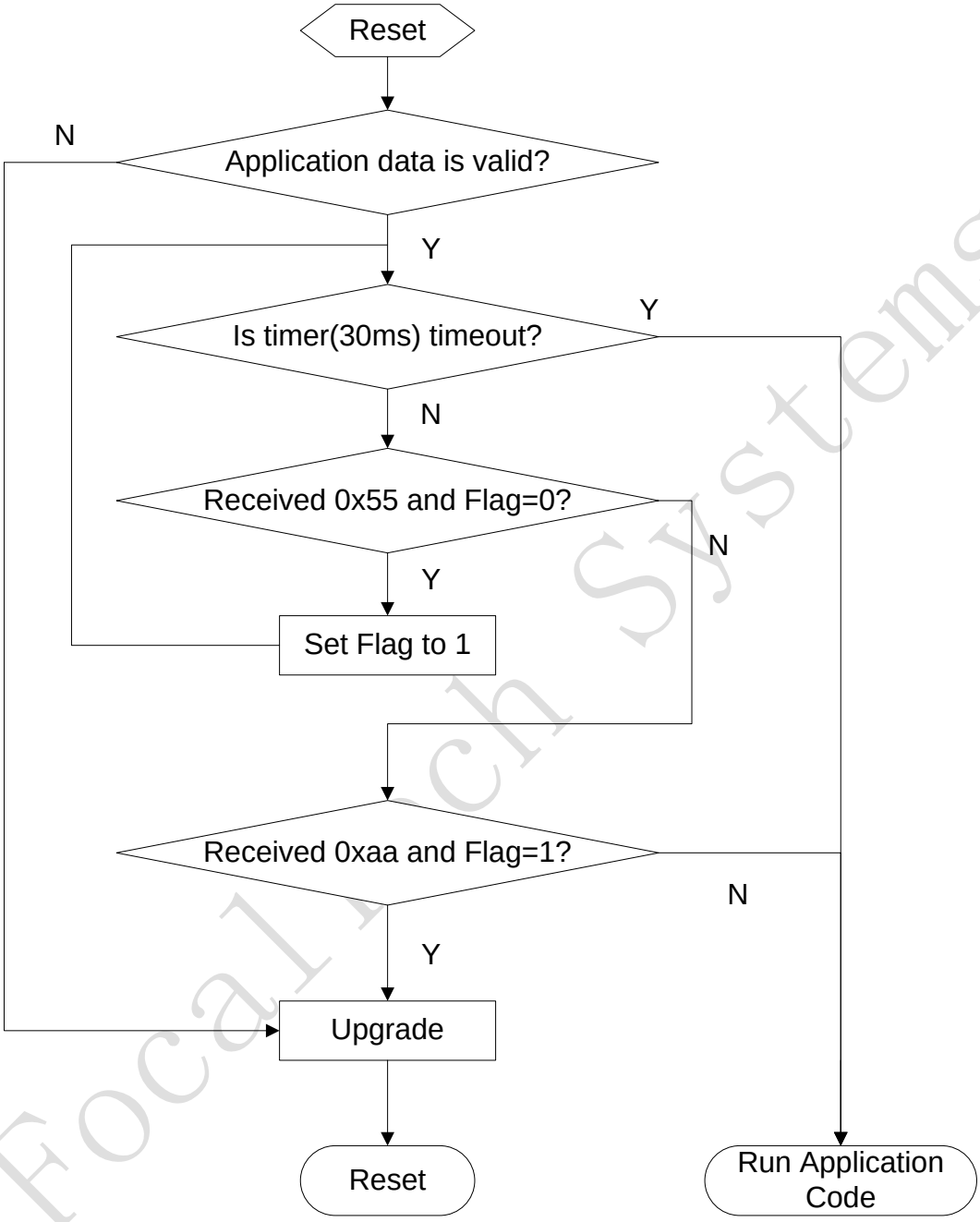


Figure 3-1: The flow chart of active upgrade

4 Application data verification

Application data verification will be run before upgrade flow.

4.1 Flash structure

Below table shows the flash structure of FT6416/FT6426.(48k)

	0	1	2	…	1022	1023
0K = 0x0	Bootloader					
1k = 0x400	Bootloader				FW configuration area(0x7b0~0x7ff)	
2k = 0x800	Interrupt vector address (0x800~0x8ff)			FW verification area(0x900~0x91f)	Application	
…	Application					
46k=0xb800	Application					
47k=0xbc00	Parameter configuration(0xbc00~0xbff7)					For tp tool (0xbff8~0xbfff: 8bytes)

Table 4-1: The structure of flash

4.1.1 FW verification area

0x900	0x901	0x902	0x903	0x904	0x905	0x906	0x907
Code length		Negation of code length		App Part1 ECC	Negation of App part1 ECC	App Part2 ECC	Negation of App part2 ECC
0x908	0x909	0x90a	0x90b	0x90c	0x90d	0x90e	0x90f
Panel Vendor ID	Negation of Panel Vendor ID	FW Version	Negation of FW Version	Upgarde flag1	Upgarde flag2	Upgarde times	Negation of Upgarde times
0x910	0x911	0x912	0x913	0x914	…	…	0x91f
the parameter region length		Negation of the parameter region length		Reserved	….	….	Reserved

Address is at Flash struct.

Code length is App F/W length.

App Part1 ECC is ECC of App F/W first 256 bytes.

App Part2 ECC is ECC of App F/W of starting at address 0x120. (Address at Flash structure is 0x920)

4.1.2 FW configuration area

0x7b0	0x7b1	0x7b2	0x7b3	0x7b4	0x7b5	...	0x7cf
IIC	Negation of	iovoltage	Negation of	Panel Vendor ID	Negation of Panel	Reserved	Reserved

Slave_address	Slave_address		Iovoltage		Vendor ID		
0x7d0	...						0x7ef
Project Code(ASCII)							
0x7f0	...						0x7ff
Customer code(ASCII)							

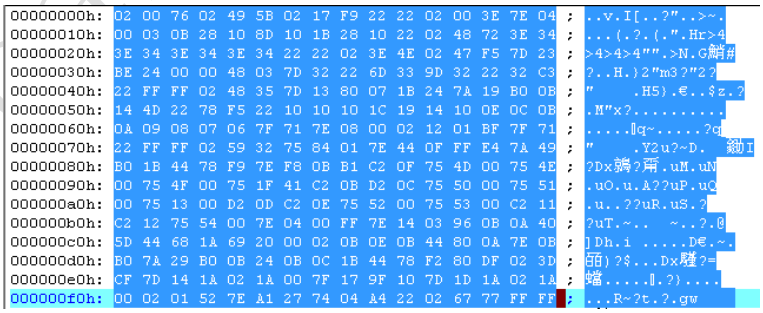
Project code and Customer code end with ASCII character '/0'.

4.1.3 Parameter configuration area

0xbc00-0xbc01	0xbc02-0xbc03	0xbc04-0xbc05	0xbc06-0xbc07	0xbc08-0xbc09	0xbc0a	...	0xbc0f
Parameter length	Negation of Parameter length	Parameter data ECC	Negation of Parameter data ECC	Start address of Parameter data	xml version	Reserved	Reserved
0xbc10	...						0xbc1f
Parameter data or 0xff							
.....							
0xbff0	...						0xbfff
Parameter data or 0xff				(0xbff8~0xbfff: 8bytes)			

4.2 App.bin file structure

"app.bin" and "app.i" is the same thing with different file format. The "app.bin" is Binary data and "app.i" is Hex data.




```
0x2, 0x0, 0x76,0x2, 0x49,0x5b,0x2, 0x17,0xf9,0x22,0x22,0x2, 0x0, 0x3e,0x7e,0x4,
0x0, 0x3, 0xb, 0x28,0x10,0x8d,0x10,0x1b,0x28,0x10,0x22,0x2, 0x48,0x72,0x3e,0x34,
0x3e,0x34,0x3e,0x34,0x3e,0x34,0x22,0x22,0x2, 0x3e,0x4e,0x2, 0x47,0xf5,0x7d,0x23,
0xbe,0x24,0x0, 0x0, 0x48,0x3, 0x7d,0x32,0x22,0x6d,0x33,0x9d,0x32,0x22,0x32,0xc3,
0x22,0xff,0xff,0x2, 0x48,0x35,0x7d,0x13,0x80,0x7, 0x1b,0x24,0x7a,0x19,0xb0,0xb,
0x14,0x4d,0x22,0x78,0xf5,0x22,0x10,0x10,0x10,0x1c,0x19,0x14,0x10,0xe, 0xc, 0xb,
0xa, 0x9, 0x8, 0x7, 0x6, 0x7f,0x71,0x7e,0x8, 0x0, 0x2, 0x12,0x1, 0xbf,0x7f,0x71,
0x22,0xff,0xff,0x2, 0x59,0x32,0x75,0x84,0x1, 0x7e,0x44,0xf, 0xff,0xe4,0x7a,0x49,
0xb0,0x1b,0x44,0x78,0xf9,0x7e,0xf8,0xb, 0xb1,0xc2,0xf, 0x75,0x4d,0x0, 0x75,0x4e,
0x0, 0x75,0x4f,0x0, 0x75,0x1f,0x41,0xc2,0xb, 0xd2,0xc, 0x75,0x50,0x0, 0x75,0x51,
0x0, 0x75,0x13,0x0, 0xd2,0xd, 0xc2,0xe, 0x75,0x52,0x0, 0x75,0x53,0x0, 0xc2,0x11,
0xc2,0x12,0x75,0x54,0x0, 0x7e,0x4, 0x0, 0xff,0x7e,0x14,0x3, 0x96,0xb, 0xa, 0x40,
0x5d,0x44,0x68,0x1a,0x69,0x20,0x0, 0x2, 0xb, 0xe, 0xb, 0x44,0x80,0xa, 0x7e,0xb,
0xb0,0x7a,0x29,0xb0,0xb, 0x24,0xb, 0xc, 0x1b,0x44,0x78,0xf2,0x80,0xdf,0x2, 0x3d,
0xcf,0x7d,0x14,0x1a,0x2, 0x1a,0x0, 0x7f,0x17,0x9f,0x10,0x7d,0x1d,0x1a,0x2, 0x1a,
0x0, 0x2, 0x1, 0x52,0x7e,0xa1,0x27,0x74,0x4, 0xa4,0x22,0x2, 0x67,0x77,0xff,0xff,
```

Example for app.bin and app.i

Below table shows the structure of "app.bin" file.

	0	1	2	…	1022	1023
0k = 0x0	Interrupt vector address (0x000~0xff)			FW verification area(0x100~0x11f)	Application	
1k=0x400	Application					
……	Application					
	Application or 0xff					
45k=0xb400	Application or 0xff (address: ~ 0xb7f7)				FW configuration area (address: 0xb7f8~0xb847)	
46k=0xb800 ~0xb847	FW configuration area (address: 0xb7f8~0xb847, 80bytes)					

The data from 0x100 to 0x11f at app.bin is totally the same with the data from 0x900 to 0x91f at flash.

0x100	0x101	0x102	0x103	0x104	0x105	0x106	0x107
Code length		Negation of code length		App Part1 ECC	Negation of App part1 ECC	App Part2 ECC	Negation of App part2 ECC
0x108	0x109	0x10a	0x10b	0x10c	0x10d	0x10e	0x10f
Panel Vendor ID	Negation of Panel Vendor ID	FW Version	Negation of FW Version	Upgarde flag1	Upgarde flag2	Upgarde times	Negation of Upgarde times
0x110	0x111	0x112	0x113	0x114	…	…	0x11f
the parameter region length		Negation of the parameter region length		Reserved	….	….	Reserved

The data from "APP_CODE_LEN+0x00" to "APP_CODE_LEN+0x4f" at app.bin is totally the same with the data from 0x7b0 to 0x7ff at flash.

APP_CODE_LEN = ((data@0x100) << 8) + (data@0x101).

APP_CODE_ LEN+0x00	APP_CODE_ LEN+0x01	APP_CODE _LEN+0x02	APP_CODE_ LEN+0x03	APP_CODE _LEN+0x04	APP_CODE_ LEN+0x05	…	APP_CODE _LEN+0x1f
IIC Slave_address	Negation of Slave_address	iovoltage	Negation of Iovoltage	Panel Vendor ID	Negation of Panel Vendor ID	Reserved	Reserved
APP_CODE_ LEN+0x20	…						APP_CODE _LEN+0x3f
Project Code(ASCII)							
APP_CODE_ LEN+0x40	…						APP_CODE _LEN+0x4f
Customer code(ASCII)							

4.3 Application verification flow

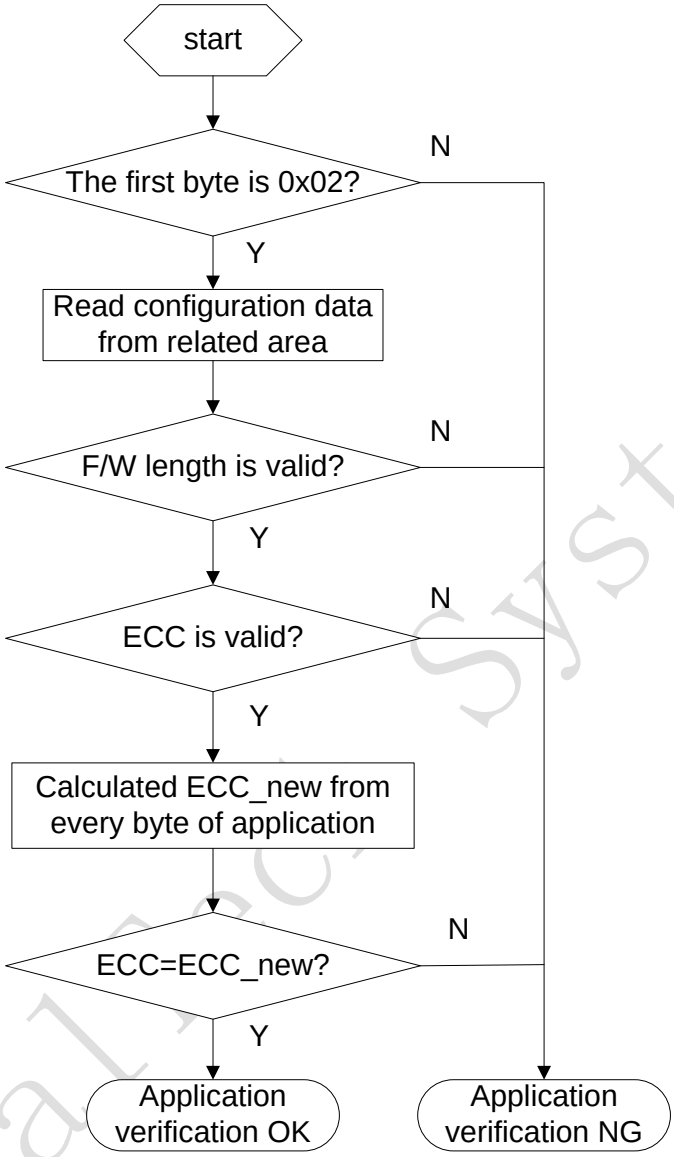


Figure 4-2: The flow chart of application verification

5 Communication protocol

Definition：

Slv_addr -------- IIC slave address

R -------- Stand for value 1，IIC read

W --------- Stand for value 0，IIC write

5.1 Communication steps

5.1.1 Read ID

Read ID is a must be step before upgrade, upgrade can't start if the ID is not correct, the ID of FT6416/FT6426 is "0x79, 0x18"

Step 1：Send the command bytes

Start	Slv_addr + W	0x90	0x00	0x00	0x00	Stop

Step 2：Read ID

Start	Slv_addr + R	**0x79**	**0x1c**	Stop

5.1.2 Erase

5.1.2.1 Erase application area

Start	Slv_addr + W	0x61	Stop

It takes ****ms [****ms (erase time for 60 blocks) + ****ms (safe delay after last erase finished)] to erase application area (46K Bytes). addr:0x0800->0xbfff.

5.1.2.2 Erase parameter configuration area

Start	Slv_addr + W	0x62	Stop

It erase config only addr: 0x b400->0xb7ff.

5.1.2.3 Erase application area A

Start	Slv_addr + W	0x63	Stop

It erase app only addr:0x0800->0xbbff.

5.1.2.4 Enable/Disable flash operation status INT

Start	Slv_addr + W	0x6b	0x00	0x00	0x00	EnINT	Stop

EnINT: 0x01->Enabled INT signal for flash operation status query;

0x00->Disabled INT signal for flash operation status query;

5.1.2.5 Read Erase finished status

Step1:
If flash operation status INT had been enabled, wait for the INT signal or timeout.

Step2:

Start	Slv_addr + W	0x6a	0x00	0x00	0x00	Stop

Step3:

Start	Slv_addr + R	Flash_Operation_Status _High_Bytes	Flash_Operation_Status _Low_Bytes	Stop

Flash_Operation_Status: 0xB002 means flash erase is finished. Other value means not yet finished and need to wait then query again.

5.1.3 Application F/W Data write

5.1.3.1 Send data package

Start	Slv_addr + W	0xBF	Add_H	Add_M	Add_L	Len_H	Len_L	Data1	...	Data	Stop

Parameter	Description	Default
Add_H	The high byte of start address	1 Byte
Add_M	The middle byte of start address	1Byte
Add_L	The low byte of start address	1Byte
Len_H	The high byte of sending data length	1Byte
Len_L	The low byte of sending data length	1Byte

Table 5-1 Abbreviation

The length of sending data can NOT exceed 255 Bytes.
Host could divided whole application data into many data packages based on the total size of application.

5.1.3.2 Read Write Flash status

Step1:
If flash operation status INT had been enabled, wait for the INT signal or timeout.

Step2:

Start	Slv_addr + W	0x6a	0x00	0x00	0x00	Stop

Step3:

Start	Slv_addr + R	Flash_Operation_Status _High_Bytes	Flash_Operation_Status _Low_Bytes	Stop

Flash_Operation_Status: (0xB002 + i) means flash write for the No. i package(128 bytes) is finished.
(1 ≦ i ≦ 0xFFD)

5.1.4 Verification

Step 1：

Start	Slv_addr + W	0xcc	Stop

Step 2：

Start	Slv_addr + R	Ecc	Stop

Ecc is the block check character of all the valid F/W bytes.
Host will also calculate ECC data by itself when sending data. It should compare the ECC data read from FT6416/FT6426 and the one which calculated by itself. If they are the same, it means that upgraded succeed, otherwise upgrade is failed and need to retry from chapter 5.1.2.

5.1.5 Read

Host could read data from FT6416/FT6426
Step 1：

Start	Slv_addr + W	0x03	ADD_H	ADD_M	ADD_L	Stop

Step 2：

Start	Slv_addr + R	Data1	Data2	……	DataN	Stop

Parameter	Description	Default
Add_H	The high byte of start address	1 Byte
Add_M	The middle byte of start address	1Byte
Add_L	The low byte of start address	1 Byte
Data	Application data	

Table 5-2 Abbreviation

5.1.6 Reset

Send this command could make FT6416/FT6426 trigger software reset.

Start	Slv_addr + W	0x07	Stop

5.1.7 Read Configuration

Step 1：

Start	Slv_addr + W	0x4A	0x00	0x00	Offset	Stop

Step 2：

Start	Slv_addr + R	Data1	Data2	Data3	…	DataN	Stop

Offset: Configuration Data Address offset (Max 0x4f).

Data: Configuration Data((N + offset) ≦ 0x50).

5.1.8 Read bootloader version

Step 1：

Start	Slv_addr + W	0xcd	Stop

Step 2：

Start	Slv_addr + R	Bootloader Version	Stop

6 Implementation

6.1 Basic Flow

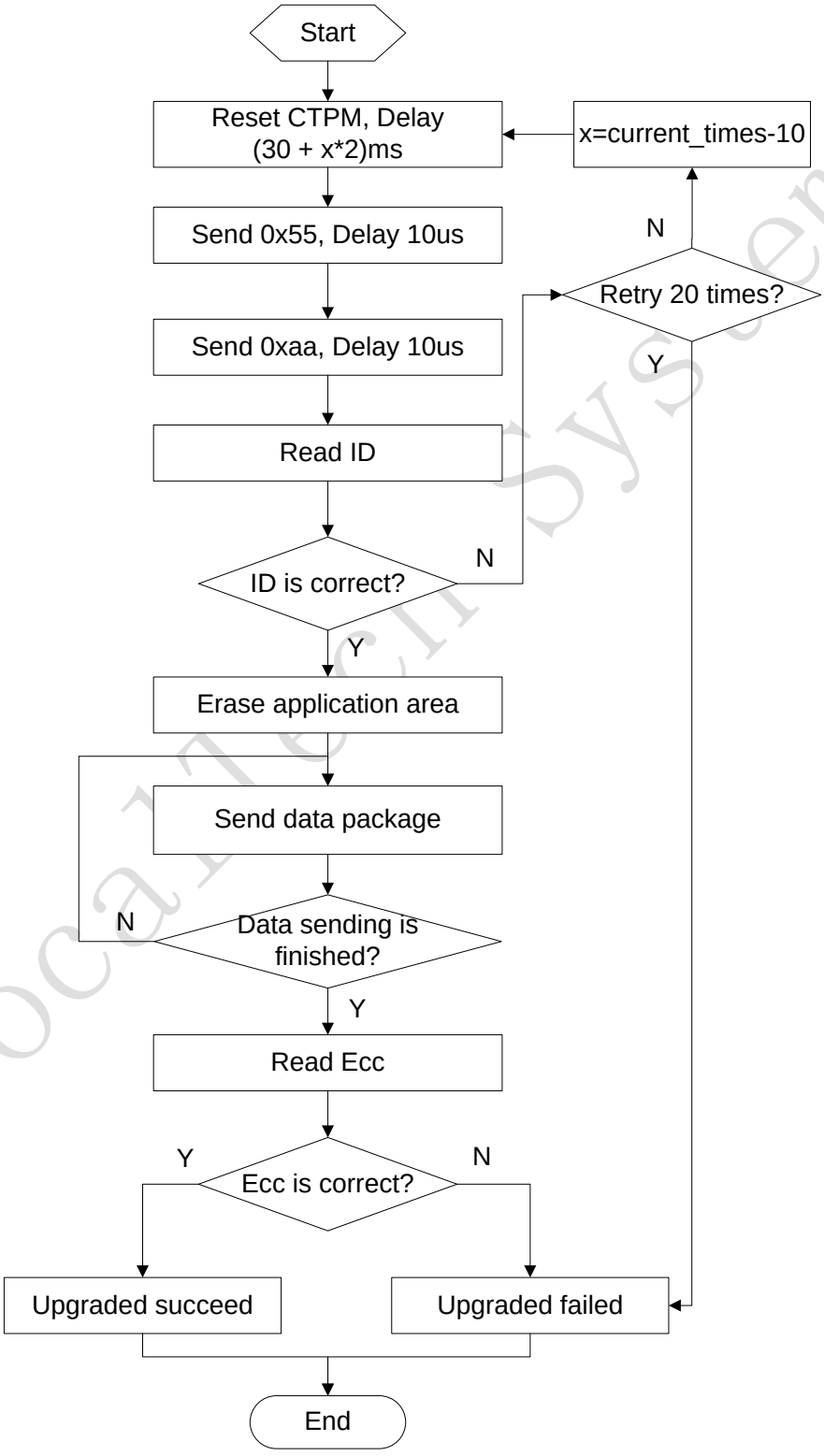


Figure 6-1: The flow chart of upgrade

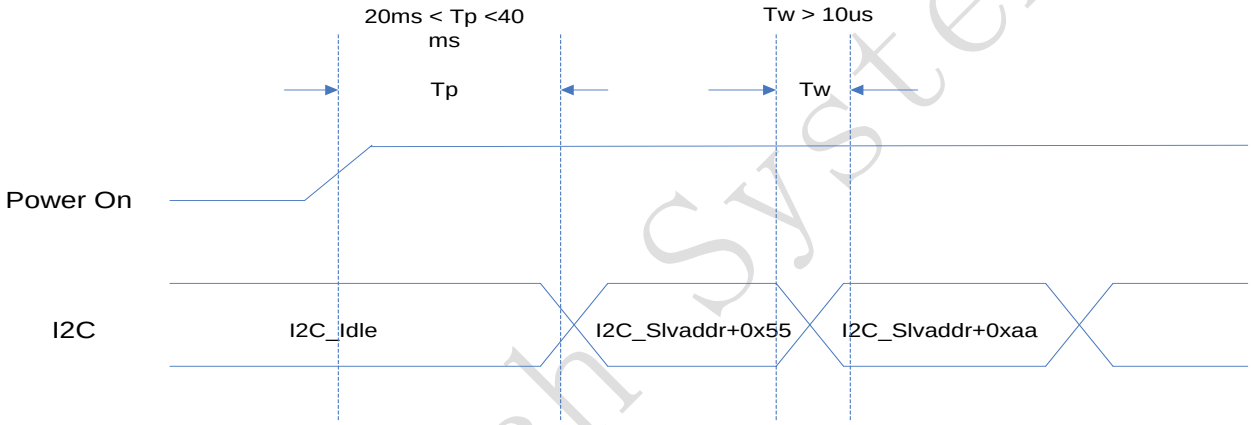
注：1、 In “Erase application area” step, if Host have not received INT after 20s, Host Automate for the next step.

2、In “Send data package” step, if Host have not received INT after 80ms, Host Automate for the next step.

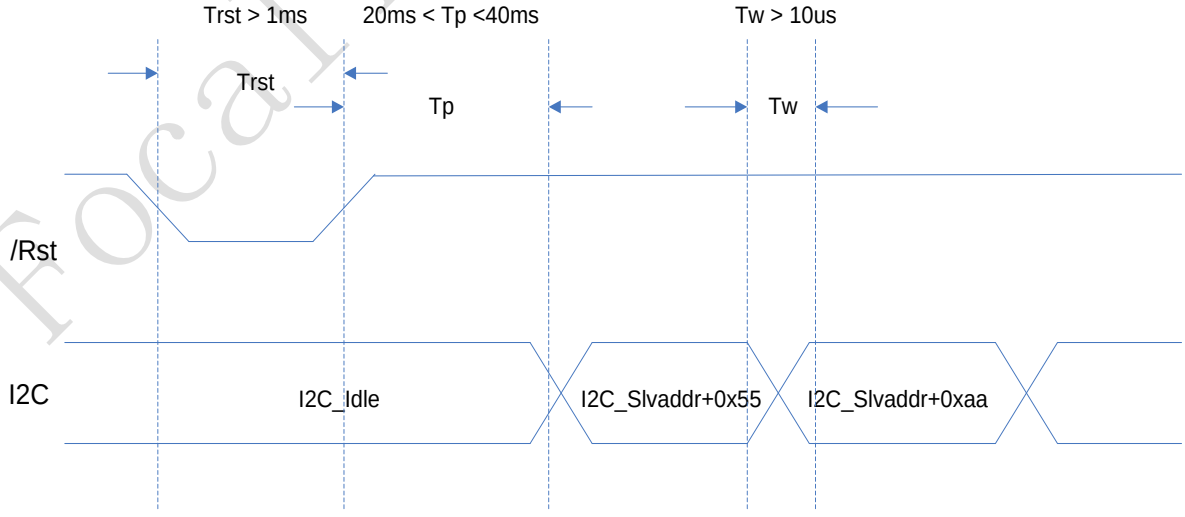
6.2 Basic Sequence

There are two methods to trigger upgrade, one is power on and the other is reset.

6.2.1 Power on Sequence



6.2.2 Reset Sequence



6.2.3 I2C Sequence