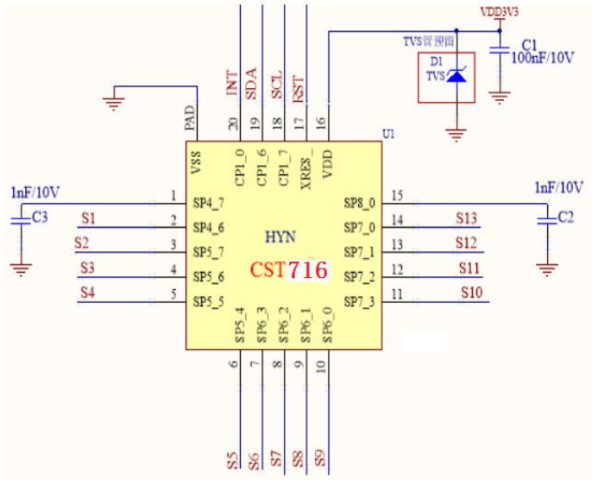
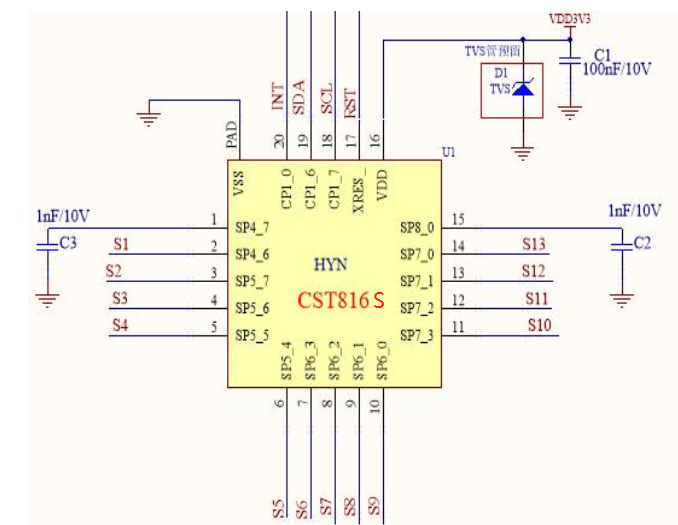
# 1. Package schematic diagram

1. cst716 and cst816s are packaged as pin to pin, which can be directly replaced.

Detailed design parameters, check design requirements, data sheet



# Two, debugging matters needing attention

1. Before debugging the project, you should contact the agent business and let the business send project debugging emails.
2. Each project has a corresponding TP firmware, communication protocol (to communicate with the main control).
3. The main board of the project, different TP manufacturers, different functional films, cover thickness and other TP-related items must be re-debugged and approved.
4. When the touch IC leaves the factory, it is blank without firmware, chip ID, and IIC address. Only BootLoader needs to be debugged according to the corresponding project, or the main control can be used after upgrading the TP firmware online according to the upgrade file we gave.

# Three, touch IC and master control interaction process

1. Process

Power on🡪Touch TP🡪Touch IC pull interrupt🡪The master control following interruption reads the coordinate data through the IIC.

# Drive debugging

## 1. Touch initialization



## 2. Upgrade and debug

For COB projects, touch soldering is on the motherboard, it is not easy to update the touch IC firmware through the burning tool, so the master needs to add and upgrade the TP firmware driver. After we debug the TP firmware, upgrade the .h or .Bin upgrade files through the tool. The main controller incorporates the upgrade file into the driver, reads and writes through IIC, and burns the data into the touch IC.

note: Cst716 and cst816S upgrade driver difference: the bootmode command is different, the others are the same.

Cst716: 0xA001 register write 0xAA, Cst816s: 0xA001 register write 0xAB,

Upgrade process:

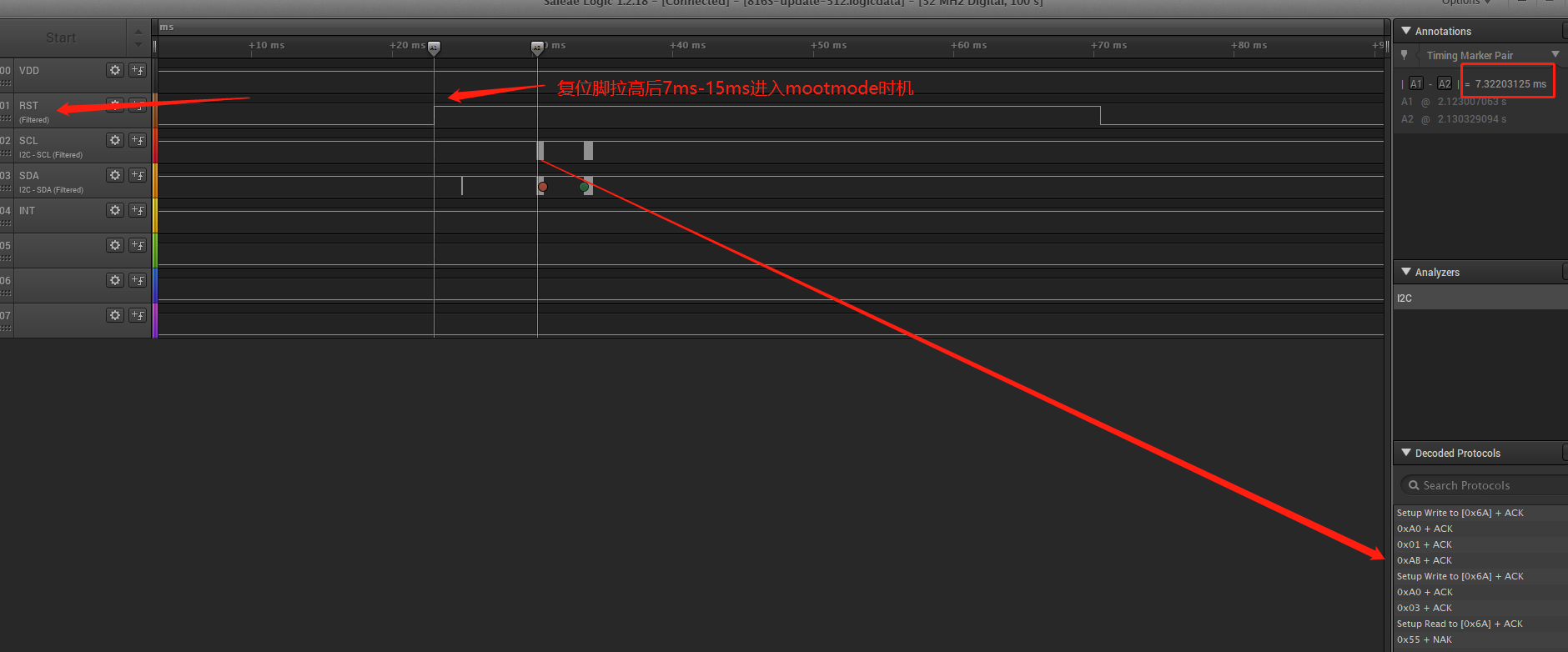
1. Enter bootmode
2. Read the checksum of the upgrade file and compare it with the checksum in the touch IC.
3. The two checksums are not the same, call the update firmware function.
4. Read the IC checksum after the upgrade. If the upgrade is successful, the checksum in the IC is the same as the checksum in the upgrade file.

* Upgrade function ctp\_hynitron\_update()



* Enter the cst816s\_enter\_bootmode() function, the IIC address is 0x6A (7 bits)





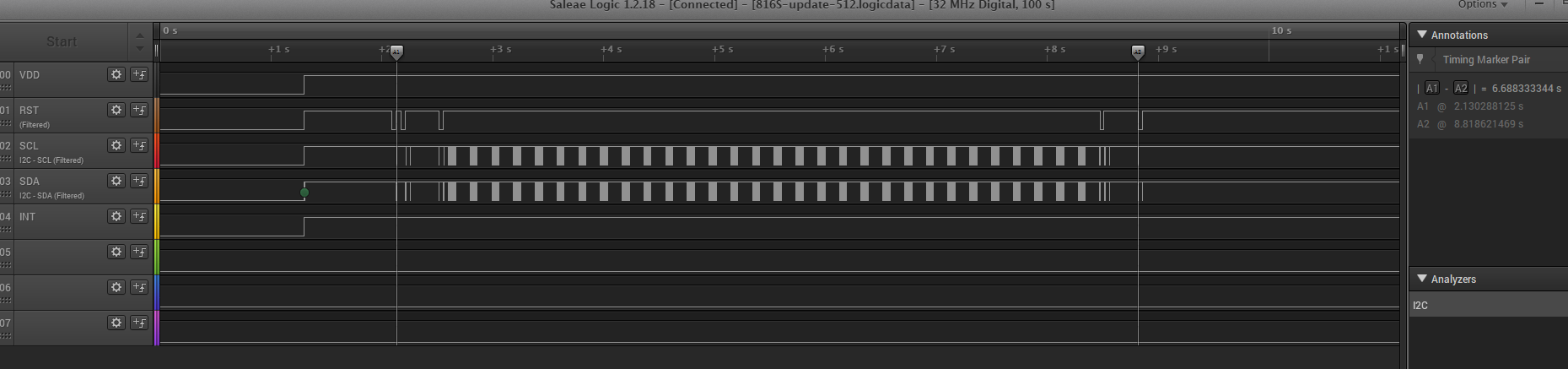
* Read cst816s\_read\_checksum() function



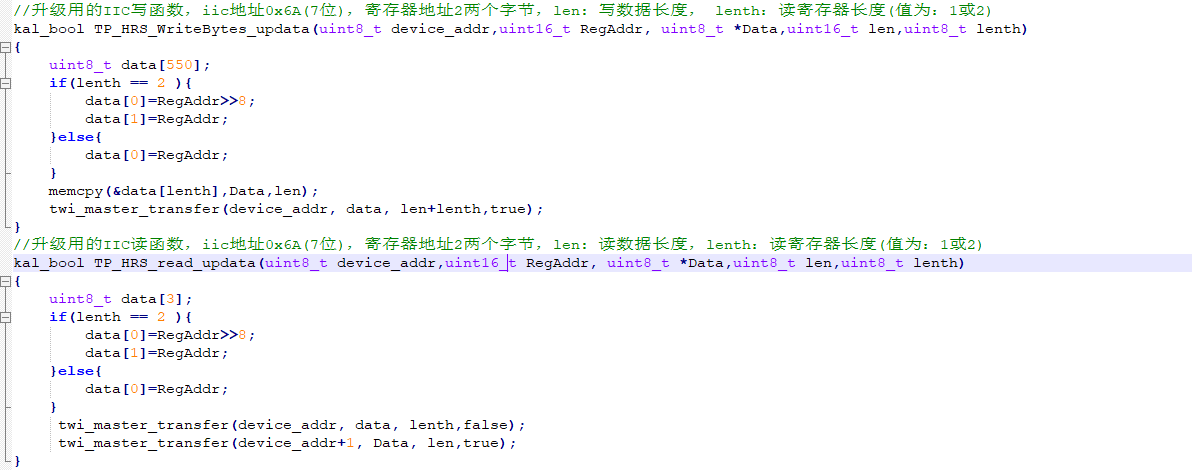
* Update data function: cst816s\_update(startAddr, length, &app\_bin[6])



Upgrade complete waveform



Note: IC upgrade needs to encapsulate two IIC read and write functions.



## 3. Sleep wake

* Sleep

cst716: The master sends a sleep command to the touch IC before the screen is off (0xA5, 0x03)

cst816S: Command mode: the master sends a sleep command to the touch IC before the screen is off (0xE5, 0x03)

Automatically enter low power consumption mode: automatically enter sleep mode without touching

* wake

cst716: The main controller wakes up the touch IC before turning on the button to turn on the screen.

cst816S: Command mode: The main controller wakes up the touch IC before turning on the button to turn on the screen. Automatically enter low power consumption mode: touch TP to wake up automatically.

Cst816s pay attention when issuing instructions: Cst816s has an automatic sleep function. When there is no touch, it will enter the low-power sleep mode. In the low-power sleep mode, the iic of the touch IC will stop working. At this time, the command IC will not respond, causing the command to fail. To ensure that the command is sent successfully, before sending the command to the touch IC, the master needs to pull the reset to wake up the touch IC.

E.g:

gpio\_pin\_write(TP\_RESET\_PIN,0); //Pull the reset pin down

delay\_ms(10); //delay 10ms

gpio\_pin\_write(TP\_RESET\_PIN,1); //Pull the reset pin up

delay\_ms(50); //It takes more than 30ms initialization time after IC reset to work normally

drv\_Touch\_Write(0xFE, 0x01, 1); //Send command, write 0x01 to 0xFE register

//Touch the IC to enter sleep mode

static void ctp\_cst816s\_deep\_sleep(void)

{

u8 data[1];

u8 regAddr;

regAddr = 0xE5; //CST816s

//regAddr = 0xA5; //cst716

data[0] = 0x03;

drv\_Touch\_Write(0xE5, data, 1); //Write 0x03 to the 0xE5 register

}

//cst816S and cst716 pull reset wake up

static void ctp\_cst816s\_wakeup(void)

{

ctp\_cst816s\_reset(); //The reset pin is pulled down for 10ms, and the delay is 50ms after being pulled up

}

## 4. Interrupt handling function

#### 4.1. There are two modes for touch IC to report data

1. Report mode

As long as there is a touch, the touch IC will pull an interrupt every 10ms or so, and the master will read the real-time coordinates according to the interrupt. Swipe up, down, left, and right, the main controller will judge the continuous coordinate value according to the reading. There is a status code for pressing and lifting.

1. Gesture mode

The touch IC is also recognized as a designated gesture code (click, double tap, long press, slide up and down, left and right) according to the user's touch track. When the gesture code is generated, it will pull an interrupt, and the master will read the generated gesture code according to the interrupt.

* be careful

Point reporting mode: The main control chip is required to have high-speed processing capabilities. To make the interface follow effect, you must touch the ic to set the point reporting mode.

Gesture mode: Can not do interface follow effect. If the status code is not pressed and lifted, the UI icon cannot be pressed and lifted to change the real effect (the click gesture is generated after lifting, and the status code is not pressed).

Cst816S supports point reporting mode and gesture mode. Support register configuration switch two modes.

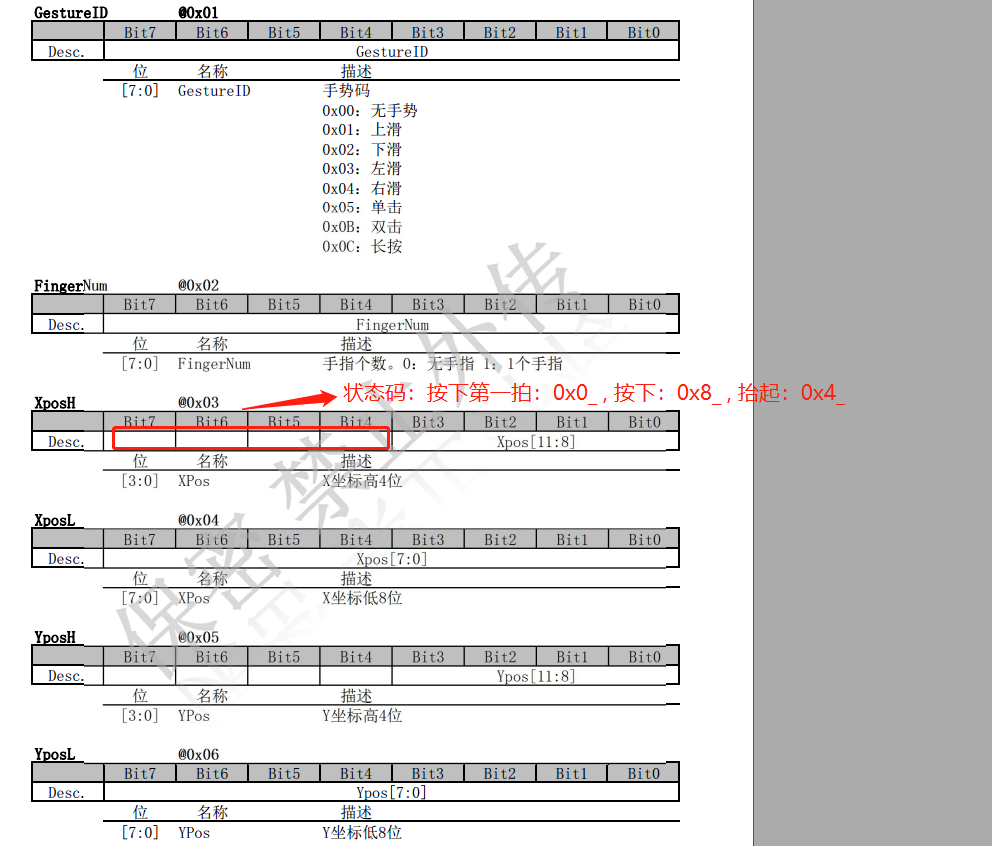
Cst716 supports the reporting mode by default, and the gesture mode needs to be communicated with FAE for modification. The two modes of register configuration switching are not supported.

#### 4.2, reporting register

After the touch IC pulls the interrupt, the master reads the data in the interrupt processing function. It is recommended to read 7 bytes continuously from the 0 register.

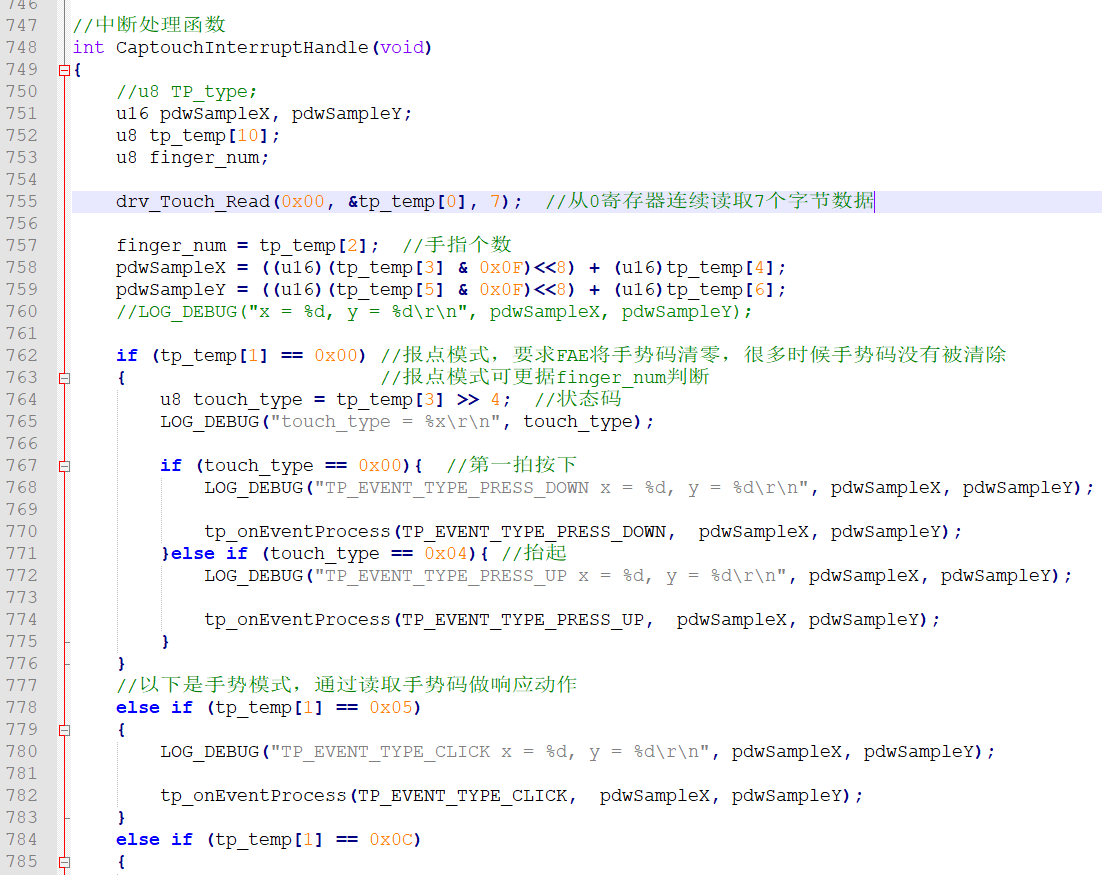
u8 tp\_temp[10];

drv\_Touch\_Read(0x00, &tp\_temp[0], 7);



#### 4.3. Analyze report point data

Analysis one



Analysis two

#### 4.4, CST816s common register configuration

Read shipID

drv\_Touch\_Read(0xA7, &chip\_id, 1); //Value cst816s: 0xB4 cst716: 0x20

Read TP firmware version number

drv\_Touch\_Read(0xA9, &g\_tp\_version, 1); //Each firmware update, the version number is incremented by 1

Set report rate, pull interrupt time interval

data[0] = 0x02; //Unit 10ms, setting time = data[0]\*10ms

drv\_Touch\_Write(0xEE, data, 1); //The default reporting rate is 100HZ, with an interrupt in 10ms,

Set to report mode

data[0] = 0x60;

drv\_Touch\_Write(0xFA, data, 1); // Pull an interrupt every 10ms or so when touching TP. Report coordinates from time to time

Set to gesture mode

data[0] = 0x11;

drv\_Touch\_Write(0xFA, data, 1); // Pull an interrupt after the gesture is generated.

Set to report point + gesture mode

data[0] = 0x71;

drv\_Touch\_Write(0xFB, data, 1);

Set auto reset time

data[0] = 0x05; //The unit is 1S. When it is 0, this function is not enabled. The default is 5

drv\_Touch\_Write(0xFC, data, 1); // When there is a touch within x seconds but there is no valid gesture, it will automatically reset

data[0] = 0x10; //The unit is 1S. When it is 0, this function is not enabled. The default is 10

drv\_Touch\_Write(0xFD, data, 1); // Automatically reset after pressing for x seconds

Prohibit automatic entry into low power mode

data[0] = 0x01; //When it is a non-zero value, it is forbidden to automatically enter the low power consumption mode.

drv\_Touch\_Write(0xFE, data, 1); // The default is 0, enable automatic entry into low power mode