CST3240 Application Manual-1.0

1、Overview

This document primarily outlines the framework architecture for the application of Haiyuechuang Touch CST3240 chip, facilitating FAE colleagues and solution companies in debugging and porting drivers. It includes parsing of the chip's touchpoint reporting protocol, firmware upgrade logic, and hardware design descriptions.

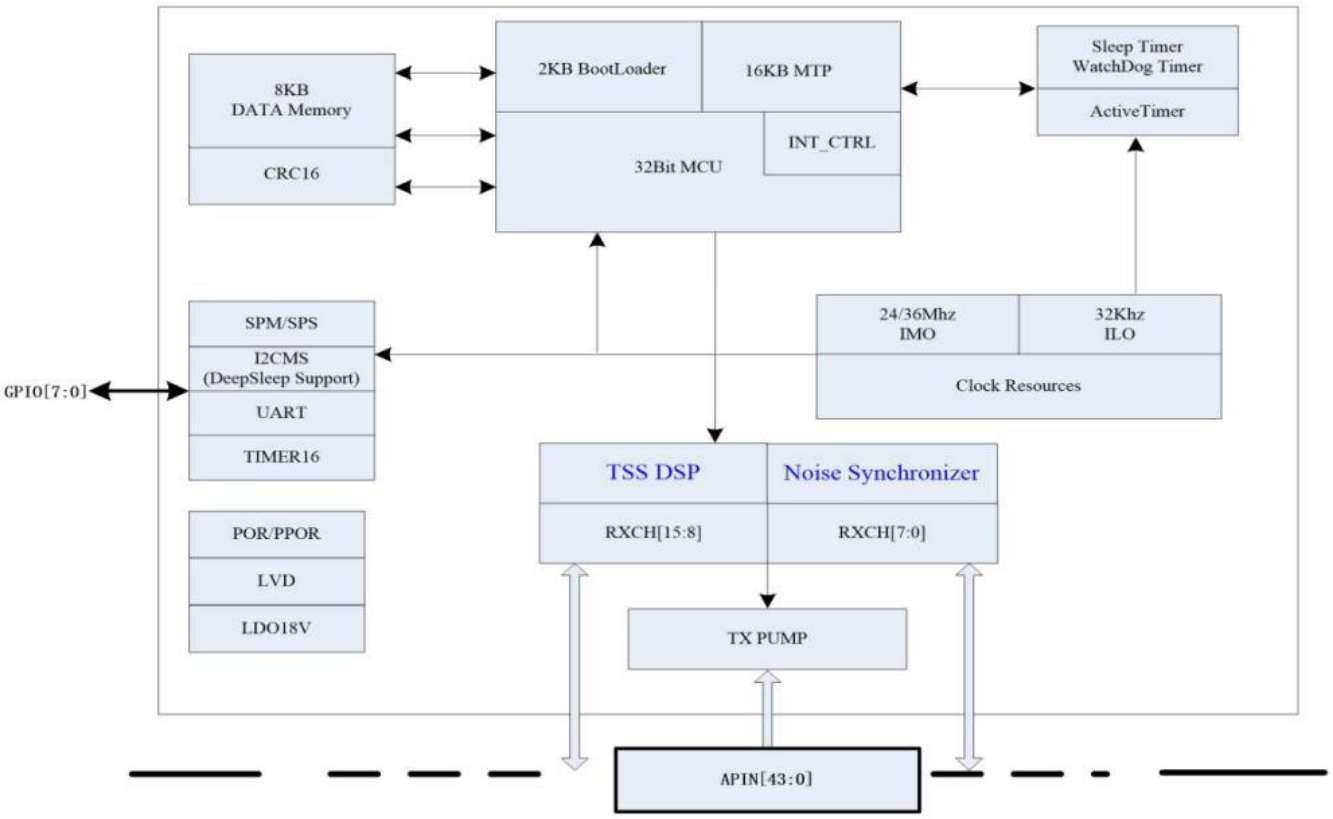
2、CST3240 Chip Introduction

2.1 Chip Hardware System Architecture

- ✧ 32-bit CM0 core, operating at 20MHz (with 3V power supply) or 32MHz (with 5V power supply)
- ✧ 16 RX detection channels
- ✧ 44 APIN pins, with TX and RX pins interchangeable
- ✧ Configurable pump voltage ranging from 4V to 7.5V
- ✧ 8KB SRAM, 24KB Romcode (2KB BootLoader), 16KB MTP (adjustable)
- ✧ Configurable VDDIO: LDO18 or VDD3V

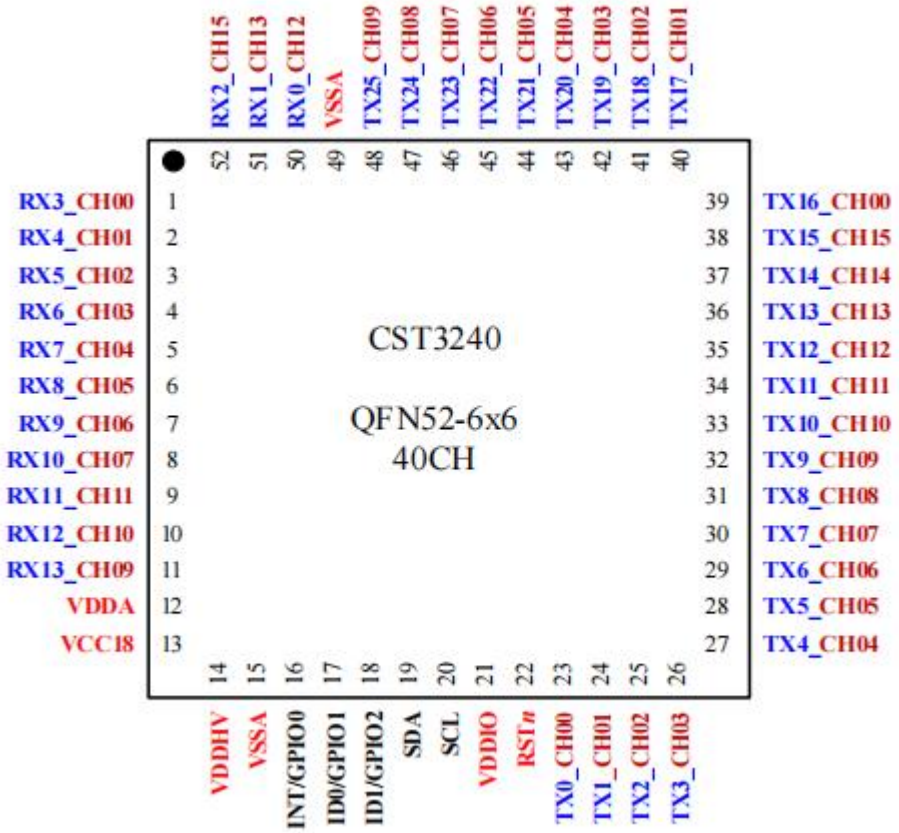
2.2 Main Features

- ✧ Self-Capacitance Waterproof/Non-Waterproof Scanning Function
- ✧ Mutual Capacitance Single-TX, MTTX Scanning Function
- ✧ Noise Listening Function
- ✧ Frequency Hopping Function
- ✧ Low Power Functions Including Deep Sleep, Light Sleep, etc.
- ✧ I2C In-Wakeup Function in Deep Sleep State
- ✧ Communication Functions Including I2C, SPI, UART, etc.
- ✧ GPIO Input/Output Function
- ✧ Configurable IO Communication Voltages: 1.8V, VDD3, etc.
- ✧ Periodic Timer and Watchdog Function
- ✧ DSP Computation Functions (Reading CSX_DR, Filling CSX_BL, Solving MTTX, etc.


8KB
DATA Memory
CRC16
2KB BootLoader
16KB MTP
32Bit MCU
INT_CTRL
Sleep Timer
WatchDog Timer
ActiveTimer
SPM/SPS
I2CMS
(DeepSleep Support)
UART
TIMER16
GPIO[7:0]
24/36Mhz
IMO
32Khz
ILO
Clock Resources
TSS DSP
Noise Synchronizer
RXCH[15:8]
RXCH[7:0]
POR/PPOR
LVD
LDO18V
TX PUMP
APIN[43:0]

3 、Hardware Design Description

3.1 Channel Configuration Description



(1) Basic Principles of Channel Configuration

For RX channels: When scanning both mutual capacitance and self-capacitance signals, it must be ensured that no duplicate internal RXCH channels are used when all RX sensors are scanned simultaneously.

For TX channels: When scanning the self-capacitance signals of TX sensors, if the number of TX sensors is less than or equal to 16, ensure that the scan can be completed in a single pass. If the number of TX sensors exceeds 16, they need to be divided into odd and even groups and scanned in two separate passes.

(2) Reference Methods for Channel Configuration

For RX sensor channels:

If the number of RX sensors is ≤ 14, you can directly use pins RX0 to RX13 for scanning. Alternatively, select N consecutive pins from TX0 to TX25 to scan the RX sensors.

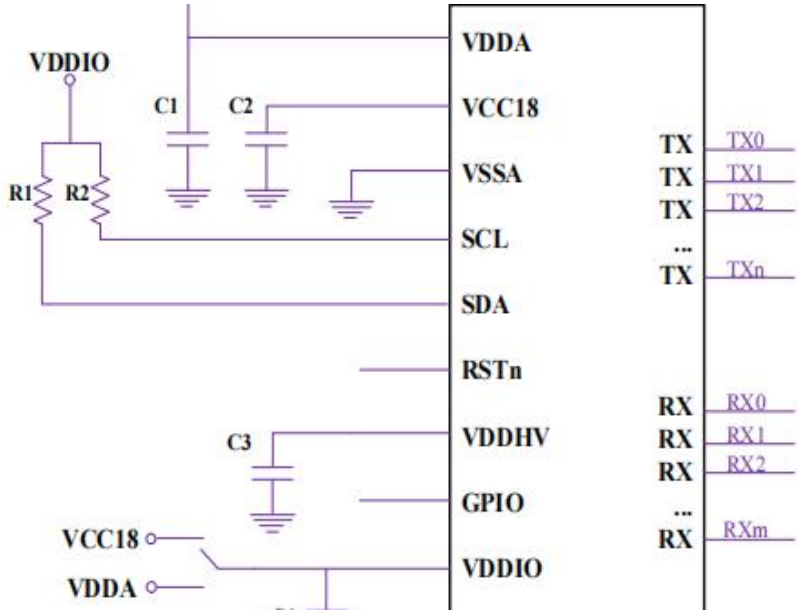
If the number of RX sensors is between 15 and 16, you can use pins RX0 to RX13 and additionally select TX8 (CH08) and TX14 (CH14) from the TX pins to scan the RX sensors. Alternatively, select N consecutive pins from TX0 to TX25 to scan the RX sensors.

For TX sensor channels:

If the number of TX sensors is ≤ 16, try to use 16 or fewer consecutive TX pins to scan the TX sensor's self-capacitance data, ensuring completion in a single scan.

If the number of TX sensors is greater than 16, when selecting TX pins, ensure that the internal RXCHn channels used for the odd-numbered TX sensor channels do not overlap, and similarly, the internal RXCHn channels used for the even-numbered TX sensor channels do not overlap. This allows the TX self-capacitance signal scan to be completed in two passes (odd and even groups).

3.2 Application Circuit Reference



C1: 2.2μF / 10V

C2: 1μF / 10V

C3: 100nF / 16V

C4: 1μF / 10V

R1/R2: I²C bus pull-up resistors. Alternatively, the chip's internal 5kΩ pull-up resistors can be configured for this purpose.

VDDIO: Must be connected to either VDDA or VCC18, depending on the required I²C or SPI communication voltage. If VDDIO is connected to VCC18, the LDO18 must be enabled in the firmware.。

4、Touch Point Protocol Parsing

4.1 Normal Coordinate Parsing

The CST3240 chip has a default 7-bit communication address of 0x5A and supports both interrupt and polling methods for reading touch points. It also supports key event reporting .

To read coordinate data, start from the register address 0xD000. Reading 7 bytes continuously will provide the coordinate information for the first finger, including the number of fingers and the number of keys pressed .

After reading the coordinates, a synchronization command must be issued to complete the current coordinate read operation. This is done by writing the value 0xD000AB .

Important Note: Touch information must be acquired in Normal Mode; otherwise, the data will be abnormal. To enter this mode, write the value 0xD109

0x5A W 0xD0 0x00

0x5A R 0x06 0x33 0x56 0x68 0x8F 0x01 0xAB

0x5A W 0xD0 0x00 0xAB

The touch data registers are as follows:

Register Address	High Nibble				Low Nibble			
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xD000	1st Finger ID				bit3-bit0: 1st Finger State Pressed (0x06) or Released (0x00)			
0xD001	Upper 8 bits of 1st Finger's X Coordinate X_Position >> 4							
0xD002	Upper 8 bits of 1st Finger's Y Coordinate Y_Position >> 4							
0xD003	Lower 4 bits of 1st Finger's X Coordinate X_Position & 0x0F				Lower 4 bits of the 1st finger's Y coordinate (Y_Position & 0x0F)			
0xD004	Pressure value of the 1st finger							

0xD005	Key Report Flag (0x80)	Number of Fingers Reported
0xD006	Fixed Value: 0xAB	
0xD007	2nd Finger ID	2nd Finger State: Pressed (0x06) or Released (0x00)
0xD008	Upper 8 bits of 2nd Finger's X Coordinate (X_Position >> 4)	
0xD009	Upper 8 bits of 2nd Finger's Y Coordinate (Y_Position >> 4)	
0xD00A	2nd Finger's X Coordinate (X_Position & 0x0F)	2nd Finger's Y Coordinate (Y_Position & 0x0F)
0xD00B	Pressure Value of the 2nd Finger	
0xD00C	3rd Finger ID	3rd Finger State: Pressed (0x06) or Released (0x00)
0xD00D	3rd Finger's X Coordinate (X_Position >> 4)	
0xD00E	3rd Finger's Y Coordinate (Y_Position >> 4)	
0xD00F	3rd Finger's X Coordinate (X_Position & 0x0F)	3rd Finger's Y Coordinate (Y_Position & 0x0F)
0xD010	Pressure Value of the 3rd Finger	
0xD011	4th Finger ID	4th Finger State: Pressed (0x06) or Released (0x00)
0xD012	4th Finger's X Coordinate (X_Position >> 4)	
0xD013	4th Finger's Y Coordinate (Y_Position >> 4)	
0xD014	th Finger's X Coordinate (X_Position & 0x0F)	4th Finger's Y Coordinate (Y_Position & 0x0F)
0xD015	Pressure Value of the 4th Finger	
0xD016	5th Finger ID	5th Finger State: Pressed (0x06) or Released (0x00)
0xD017	5th Finger's X Coordinate (X_Position >> 4)	
0xD018	5th Finger's Y Coordinate (Y_Position >> 4)	
0xD019	5th Finger's X Coordinate (X_Position & 0x0F)	5th Finger's Y Coordinate (Y_Position & 0x0F)
0xD01A	Pressure Value of the 5th Finger	

4.2 Physical Button Parsing

Button reporting is divided into two methods:reporting key values and reporting coordinates.The choice between these two methods is implemented in the driver.The chip provides the assigned IDs for the corresponding buttons(0x17,0x27,0x37),and the driver reports the corresponding key values or coordinates based on these IDs

```
#if report_key_value
    #define key_code        {KEY_BACK, KEY_HOMEPAGE, KEY_MENU}    // Define the required key function keycodes
#else
    / Define the required button coordinate ranges, can be modified according to actual needs
    #define key_x_coord     {200, 600, 800}
    #define key_y_coord     {2000, 2000, 2000}
```

0x5A W 0xD0 0x00

0x5A R 0x83 0x17 0x00 0x00 0x00 0x80 0xAB

0x5A W 0xD0 0x00 0xAB

Register Address	High Nibble				Low Nibble			
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xD000	The state of the key: Pressed (0x83) or Released (0x80)							
0xD001	Key ID(0x17,0x27,0x37)							
0xD002	0x00							
0xD003	0x00							
0xD004	0x00							
0xD005	Key Report Flag (0x80)							
0xD006	Fixed value: 0xAB							

The driver reads 7 bytes of data from register 0xD000 (format as above), where buf[0]represents the key press/release state, and buf[1]represents the different key IDs. Based on this ID, the corresponding key value or the required coordinates can be reported.

5、Driver Upgrade Process

Both external reset and power-on reset can cause the chip to restart and enter the BootLoader. During the initialization phase, the Bootloader verifies the main area. After completing the necessary initialization and verification (approximately 10ms), the chip enters a timeout period waiting for the host's firmware programming command. This timeout waiting period is about 15ms. If the BootLoader program receives the host's firmware update command within this 15ms, the program enters the firmware update process; otherwise, after the 15ms timeout, the program jumps to the main program. Therefore, the host should wait about 10ms after performing an external reset or power-on reset on the chip (waiting for the chip hardware and software initialization to complete). Within the chip's 15ms waiting period for the firmware update command, use the I2C bus to send the firmware update command to the chip.

The main upgrade process of the chip is divided into resetting the chip, entering BootLoader mode, writing the firmware, verifying the firmware, and exiting BootLoader. Among these steps, as described above, resetting the chip to enter BootLoader mode is the most important and the most prone to issues. It is necessary to ensure that the reset operation is effective; otherwise, it will be impossible to enter BootLoader mode for upgrading.

Specific Upgrade Process:
Upgrade Array File (Confirm the latest version with FAE): hynitron_cst3240_update.h
Upgrade File Size: #define CST3240_BIN_SIZE (31 * 512 + 480)

Step1: Reset the Chip
After reset, the chip maintains a BootLoader **mode** window period between 5ms and 15ms. Sending the command (within the cst3240_into_program_modefunction) during this period can make the chip enter programming mode for firmware updates. The first priority in the upgrade process is to ensure that this reset operation is effective; otherwise, upgrading is not possible. Note: Since the window period is not fixed, a variable time retry_timerneeds to be set for retry attempts.

```
hyn_reset_proc(5+retry_timer);
```

Step2: Enter Upgrade Mode
After sending the command 0xA001AAto enter upgrade mode, read the value of register 0xA002to determine if entering upgrade mode was successful. If the return value is 0x55...

Step3: Update Firmware
The firmware must be sent with a total length of (31 * 512 + 480) bytes. Write 512 bytes at a time, for a total of 31 writes. The final 480 bytes are written separately.

Step4: Verify Firmware Accuracy

After firmware transmission is complete, read the bin programming result from address 0xA000. If the register content is 0x00, verification is not complete. If it is 0x01, verification passed. If it is 0x02, verification failed.

If verification is determined to have passed, then read the firmware checksum from within the chip from register 0xA008 and compare it with the bin file checksum to see if the verification results are consistent.

```
// Firmware checksum within the chip
checksum = buf[0] + (buf[1]<<8) + (buf[2]<<16) + (buf[3]<<24);
// Bin file checksum
pData = hynitron_cst3240_update.h + 31 * 512 + 476; // 7 * 1024 + 512
bin_checksum = pData[0] + (pData[1]<<8) + (pData[2]<<16) + (pData[3]<<24);
```

If the checksums are inconsistent, it indicates a data transmission error. Return an error code.

Step5: Exit Upgrade Mode

Write 0xEE to register 0xA006 to inform the chip that the upgrade has ended, exit upgrade mode, then reset the chip. The upgrade is complete.

Note: If unable to enter bootloader mode during upgrade, please confirm the reset method:

1. Power-off reset
2. Reset pin reset
3. Watchdog reset

Window period for entering bootloader mode: Generally, commands sent within 5ms~20ms after resetting the chip are effective.

6、Firmware Information Acquisition

(1) Firmware information must be read in debug info mode (write 0xD101). To ensure the mode switch is successful, the mode command can be sent multiple times. If the command fails to send, incorrect firmware information data will be read.

(2) Read the information from the corresponding register addresses (for specific addresses and data storage format, refer to the table below).

(3) Return to normal mode (write 0xD109). After reading the firmware information, it is essential to switch back to normal mode; otherwise, correct coordinate information cannot be read.

Register Address	Register Description	Register (4 bytes) Content			
0xD1F4	Number of Keys, TX, RX Channels	KEY_NUM	TP_NRX	NC	TP_NTX
0xD1F8	X/Y Resolution	TP_RESY		TP_RESX	
0xD1FC	Firmware Checksum, Bootloader Time	0xCACA		BOOT_TIMER	
0xD204	Chip Type, Firmware Project ID	IC_TYPE		PROJECT_ID	
0xD208	Chip Firmware Version Number	FW_MAJOR	FW_MINOR	FW_BUILD	
0xD20C	Chip Firmware Checksum	checksum_H	checksum_H	checksum_L	checksum_L