T5 Series Touch IC Protocol Document

This document describes the communication protocol for the T5 project. The communication between the touch IC and ESP32 is done through UART.

1. Serial Communication convention

- Baud rate: 115200
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: None
- Checksum method: CRC16

CRC16: CRC checksum is the validation value from the packet header to the data content.
Polynomial: g(x) = x^16 + x^12 + x^5 + 1
Initial value: 0xFFFF
POLY: 0x1021

```
static uint16_t crc16_ccitt(uint8_t *in_data, uint8_t len)
{
    uint8_t i,k = 0;
    uint16_t CRC_acc = 0xFFFF;
    #define POLY_  0x1021
    while(len--)
    {
        CRC_acc = CRC_acc ^ (in_data[k++] << 8);

        for (i = 0; i < 8; i++)
        {
            if ((CRC_acc & 0x8000) == 0x8000)
            {
                CRC_acc = CRC_acc << 1;
                CRC_acc ^= POLY_;
            }
            else
            {
                CRC_acc = CRC_acc << 1;
            }
        }
    }
    return CRC_acc;
}
```

2. Frame Format Description

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55

Field	Length (Byte)	Explanation
Version	1	0x01
Opcode	1	Specific command type
Data Length	1	Data Length 0-64
Data	N	
Checksum	2	CRC16 checksum starting from the version byte

Explanation:

- All Data larger than one byte are transmitted in big-endian mode.
- In general, a command-response synchronization mechanism is used, where the sender expects to receive a response packet corresponding to the sent command. If the sender does not receive a correct response packet within the specified timeout, it will trigger a timeout transmission. *Note: Please refer to the "Protocol Detail" for specific communication methods.

3. Protocol Detail

1. Test command (0x00)

Explanation:

- The opcode for the test command is 0x00.
- After the chip is powered on, this command is sent at intervals of 200ms within 10 seconds to check if the communication on the other end is connected and working properly. If a response to this command is received, the sending of the test command stops.
- The test command is a bidirectional command, and the receiving party follows the "Correct response" protocol to return a response to the sender.

Command Sending: (ESP32 -> Touch IC; Touch IC -> ESP32)

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	0x00
Data Length	1	0x00
Checksum	2	CRC16 checksum starting from the version byte

2. Return command

Explanation:

- The receiving end checks the Data sent by the sending end. If the Frame Header, Version, or address (any one of them) in the Data is incorrect, no response is sent.

- If the DataFrame Header and checksum of a frame are correct, the Data of that frame is considered for processing, and a response command is expected.
- The receiving end responds to the operation by setting the Opcode to cmd | 0x80. The success or failure of the operation is reflected in the Data section.
- For return (settings or queries), the first byte of the Data section indicates whether it is a successful or exceptional response. 0x00 represents success, while other values represent exceptions.

Correct response

Explanation:

- The receiving end returns a correct response command, indicating that the corresponding operation sent by the sending end is without exceptions and can be executed successfully.
- Except for specific Data returned for query operations, the receiving end returns the correct response command to the sending end for any command that is sent without exceptions.

Receving end return

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	Corresponding command Opcode | 0x80
Data Length	1	0x01
Data	0	0x00 (correct return 0x00)
Checksum	2	CRC16 checksum starting from the version byte

Correct return: 0xaa 55 01 00|80 01 00 00 xx xx

Abnormal response

Explanation:

- The receiving end returns an command indicating an error or exception.

Receving end return

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	Corresponding command Opcode | 0x80
Data Length	1	0x01

Field	Length (Byte)	Explanation
Data	1	Abnormal return: 0x00: success 0x01: failed
Checksum	2	CRC16 checksum starting from the version byte

Abnormal return: 0xaa 55 01 00|80 01 01 00 xx xx (Unsupported command)

3 Operation command

Query Touch Firmware Version (0x01)

Command sending: ESP32 → Touch IC

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	0x01
Data Length	0	0x00
Data	0	
Checksum	2	CRC16 checksum starting from the version byte

Command sending: Touch IC → ESP32

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	0x81
Data Length	3	0x03
Data	3	For example, if the device version is 1.3.5. The device responds with 0x01, 0x03, 0x05. Byte0: OTA firmware major version number, range 1-9 Byte1: OTA firmware minor version number, range 0-9 Byte2: OTA firmware revision version number, range 0-9
Checksum	2	CRC16 checksum starting from the version byte

Report Touch Event (0x02)

Touch is not effective during OTA.
Command Sending: Touch IC → ESP32

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	0x02
Data Length	1	0x01-0x03
Data	1	0x00: Touch channel press event 0x01-0x0A: Short press and release events corresponding to touch channels 1-10 0x11-0x1A: Long press and release events corresponding to touch channels 1-10 0x0B: Event triggered when multiple touch channels are pressed 0x0C: Event triggered by left-to-right swipe 0x0D: Event triggered by right-to-left swipe 0x0E: 15-second long press event, only used for comprehensive testing and WSS testing for Amazon.
Data	1	1. If the touch event is a short press, long press, or channel press event, this byte represents the channel number of the first touch. 2. If the touch event is a left swipe or right swipe, this byte represents the high 8 bits of the touch channel.
Data	1	If the touch event is a left swipe or right swipe, this byte represents the low 8 bits of the touch channel. Example: If the event is a left swipe and touch_channel = 0x03FF, it represents that touch channels 1-10 are pressed.
Checksum	2	CRC16 checksum starting from the version byte

Reset Restart Notification (0x07)

- Command sending: Touch IC → ESP32
- Touch IC sends a reset restart notification.

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01

Field	Length (Byte)	Explanation
Opcode	1	0x07
Data Length	1	0x00
Checksum	2	CRC16 checksum starting from the version byte

Device Type Notification (0x08)

- Command sending: ESP32 → Touch IC
- Notifies the Touch IC of the current device type (80/86/120); default is 86.

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	0x08
Data Length	1	0x01
Data	1	0x00: Device type is 80 0x01: Device type is 86 0x02: Device type is 120
Checksum	2	CRC16 checksum starting from the version byte

Set Touch Sensitivity (0x09)

- Command sending: ESP32 → Touch IC
- The sensitivity value: the smaller the value, the more sensitive it is.

Field	Length (Byte)	Explanation
Frame Header	2	0xaa55
Version	1	0x01
Opcode	1	0x09
Data Length	1	0x14
Data	20	The sensitivity values corresponding to the touch channels. When the value is 0xFFFF, it means the sensitivity of the current channel is not modified. For the current T5-86 device, the default touch sensitivity for the 10 touch channels is as follows: 0x0023, 0x0012, 0x000c, 0x000c, 0x000c, 0x000f, 0x000f, 0x000f, 0x000f, 0x001c.

Field	Length (Byte)	Explanation
Checksum	2	CRC16 checksum starting from the version byte