ESK32-A4A32
4.3-inch 480×272 Capacitive Touch Screen Module User Guide

Revision: V1.00 Date: October 17, 2025

www.holtek.com

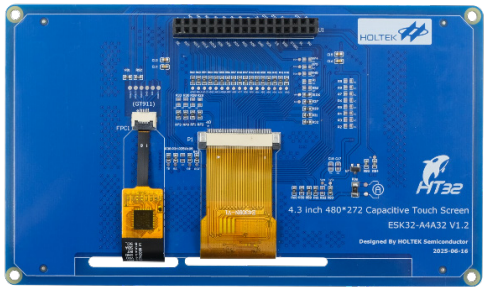
Table of Contents

1. Overview

The ESK32-A4A32 is a 4.3-inch capacitive touch screen module with a resolution of 480×272 from Holtek. Its LCD driver IC is the ST6201 with integrated RAM, eliminating the need for additional drivers or memory. It utilizes an 8080 parallel port 16-bit driver and features 16-bit color depth (RGB565 format), delivering vivid, clear and detailed images. The capacitive touch driver IC is the GT911, supporting 5-point touch simultaneously, enabling a responsive and convenient interactive experience.

This module is suitable for scenarios with fundamental requirements for display accuracy and touch interaction, such as embedded devices, industrial control terminals, smart home control panels and small medical instruments.

ESK32-A4A32 Front View



ESK32-A4A32 Back View

2. Product Parameters

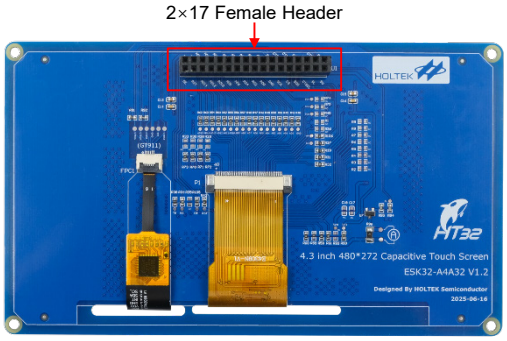
The basic parameters of the ESK32-A4A32 module are as follows:

Parameter Type	Description
LCD Communication Interface	MCU interface (8080 parallel 16-bit); display driver IC: ST6201
Display Panel	IPS
Display Color Depth Format	16-bit, RGB565
LCD Resolution	480×272 pixels
Display Outer Size	105.50mm(Width) × 67.20mm(Length) × 4.22mm(Thickness)
Display Area Size	95.04mm×53.86mm
Pixel Size	0.198mm×0.198mm
Touch Screen Communication Interface	I^2C, communication speed < 400kHz
Touch Screen Type	Capacitive; touch driver IC: GT911
Number of Touch Points	Up to 5 simultaneous touch points
Backlight Drive Voltage	LCD backlight power supply: 3.3V
I/O Port Level	3.3V
Module Operating Voltage	2.8V~3.3V
Module Size	150.0mm×88.2mm

3. Instructions for Use

3.1 Pin Description

The ESK32-A4A32 module establishes hardware connections with external circuits via a 2×17 female header (2.54mm pitch) and can be used in combination with development boards such as the ESK32-A4A11 for development purposes.



ESK32-A4A32 Female Header

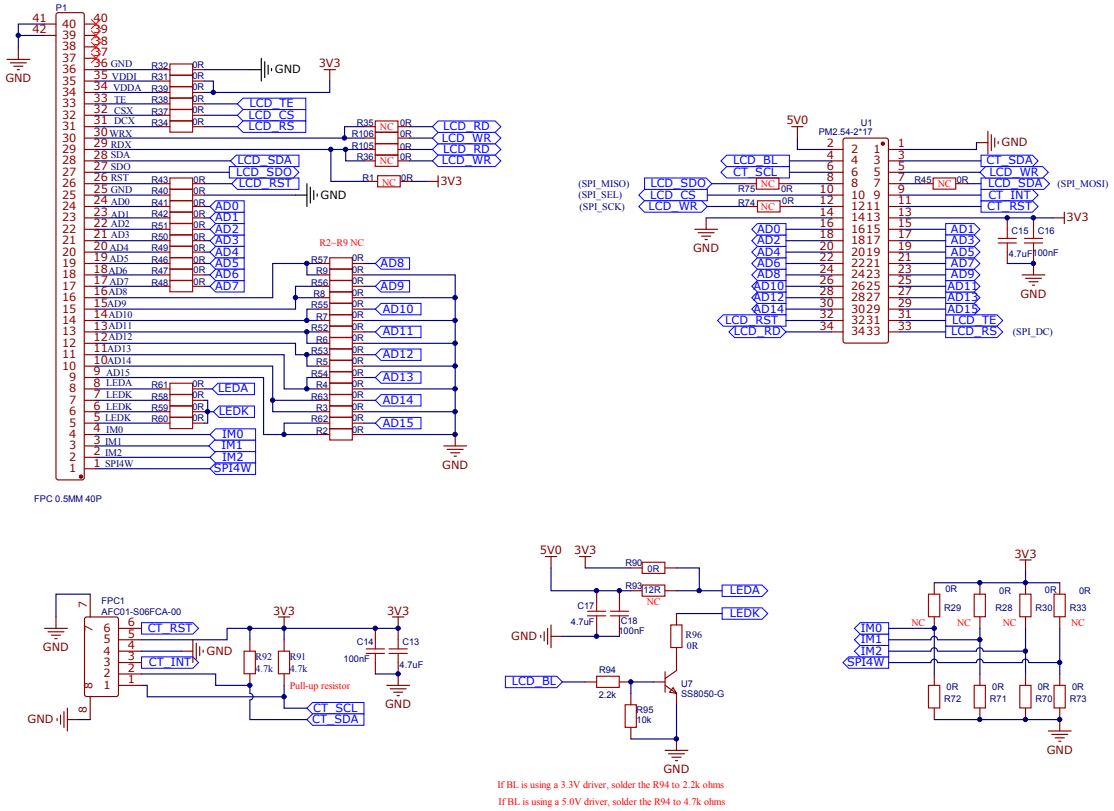
The detailed description of each pin is shown in the following table:

Pin No.	Pin Name	Functional Description	Pin No.	Pin Name	Functional Description
1	GND	Ground	2	5V	5V analog power supply
3	CTSDA	Touch screen I2C_SDA signal	4	BL	LCD backlight control, high active to turn on
5	WE	LCD 8080 interface write enable signal	6	CTSCL	Touch screen I2C_SCL signal
7	SDA	LCD SPI interface data input channel (invalid on this module)	8	SDO	LCD SPI interface data output channel (invalid on this module)
9	CTINT	Capacitive touch screen interrupt signal	10	CS	LCD driver chip select, low active
11	CTRST	Capacitive touch screen reset signal	12	SCK	LCD SPI interface clock signal (invalid on this module)
13	3V3	Module power supply (3.3V)	14	GND	Ground
15	AD1	MCU parallel data bus bit 1	16	AD0	MCU parallel data bus bit 0
17	AD3	MCU parallel data bus bit 3	18	AD2	MCU parallel data bus bit 2
19	AD5	MCU parallel data bus bit 5	20	AD4	MCU parallel data bus bit 4
21	AD7	MCU parallel data bus bit 7	22	AD6	MCU parallel data bus bit 6
23	AD9	MCU parallel data bus bit 9	24	AD8	MCU parallel data bus bit 8
25	AD11	MCU parallel data bus bit 11	26	AD10	MCU parallel data bus bit 10
27	AD13	MCU parallel data bus bit 13	28	AD12	MCU parallel data bus bit 12
29	AD15	MCU parallel data bus bit 15	30	AD14	MCU parallel data bus bit 14
31	TE	LCD frame synchronization signal	32	RST	LCD reset signal, low active
33	RS	LCD 8080 interface display command / display data select	34	OE	LCD 8080 interface read enable signal, connected to high level when not used

It is worth noting that the ST6201 controller determines its communication mode with the MCU by detecting the logic state of its own IM[3:0] signal lines. While it supports communication protocols such as SPI and 8080, during the circuit design phase this module has been fixed to operate exclusively as an 8080 16-bit interface; other drive modes are disabled. Furthermore, the LCD backlight uses a 3.3V power supply, with the 5.0V supply remaining unused.

3.2 Module Schematic

The figure illustrates the schematic of the ESK32-A4A32 module.



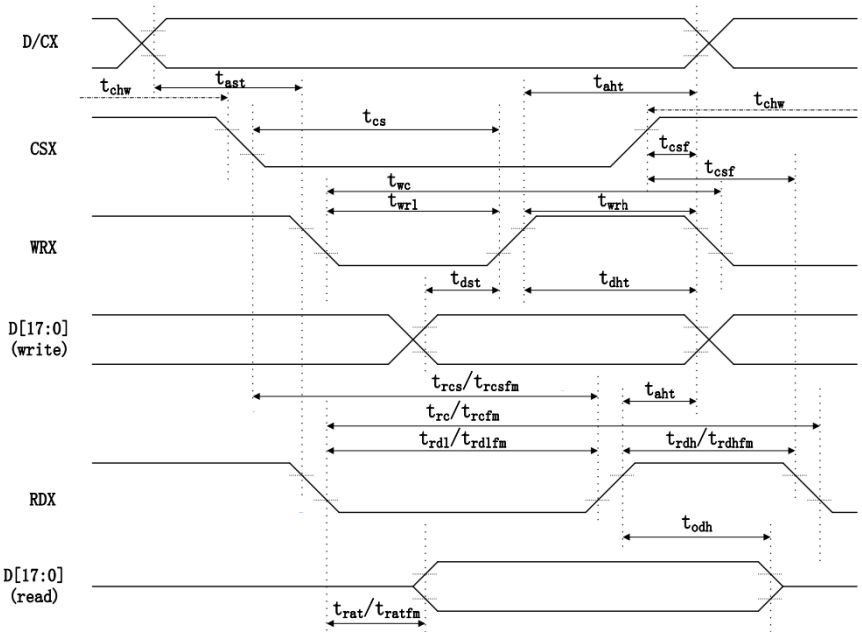
ESK32-A4A32 Schematic

Note: The Components marked with “NC” (such as R2-R9, etc.) are reserved for future use and no soldering is required in actual applications.

4. LCD Driver Introduction

The display driving of the ESK32-A4A32 module lies in controlling the ST6201 LCD chip. This process primarily involves two tasks: first, transmitting control commands — various commands specific to the ST6201; second, transmitting display memory data — the RGB565 format content corresponding to each pixel. The host can send command codes to the ST6201 via the 8080 interface to execute control functions, including reset, cursor positioning, sleep mode and other commands. Refer to the ST6201 datasheet for more detailed command specifications.

The ST6201 host driver must establish an 8080 timing mechanism to transmit commands and data to the LCD screen. The timing parameters are as follows:



ST6201 Timing Diagram

The ST6201 timing parameters are listed in the table:

Signal	Symbol	Parameter	Condition	Min.	Max.	Unit
D/CX	t_{ast}	Address Setup Time	—	0	—	ns
	t_{aht}	Address Hold Time (W/R)	—	10	—	ns
CSX	t_{chw}	Chip Select High Pulse Width	—	0	—	ns
	t_{cs}	Chip Select Setup Time (W)	—	10	—	ns
	t_{rcs}	Chip Select Setup Time (Read ID)	—	45	—	ns
	t_{rcsfm}	Chip Select Setup Time (Read FM)	—	355	—	ns
	t_{csf}	Chip Select Wait Time (W/R)	—	10	—	ns
WRX	t_{wc}	Write Cycle	—	66	—	ns
	t_{wrh}	Control Pulse High Level Duration	—	25	—	ns
	t_{wrl}	Control Pulse Low Level Duration	—	25	—	ns
RDX	t_{rc}	Read Cycle (ID)	—	160	—	ns
	t_{rdh}	Control Pulse High Level Duration (ID)	—	90	—	ns
	t_{rdl}	Control Pulse Low Level Duration (ID)	—	45	—	ns
	t_{rcfm}	Read Cycle (FM)	—	450	—	ns
	t_{rdhfm}	Control Pulse High Level Duration (FM)	—	90	—	ns
	t_{rdlfm}	Control Pulse Low Level Duration (FM)	—	355	—	ns

Signal	Symbol	Parameter	Condition	Min.	Max.	Unit
D[17:0]	t_{dst}	Data Setup Time	CLmax=30pF, CLmin=8pF	10	—	ns
	t_{dht}	Data Hold Time		10	—	ns
	t_{rat}	Read Access Time (ID)		—	40	ns
	t_{ratfm}	Read Access Time (FM)		—	340	ns
	t_{odh}	Output Disable Time		20	80	ns

As shown in the table, the minimum write cycle time (t_{wc}) is 66ns. Regarding the read cycle (t_{rc}), the minimum Read Cycle (ID) is 160ns, while the minimum Read Cycle (FM) is 450ns. Users should refer to the ST6201 timing requirements and make appropriate adjustments when configuring the drive timing.

5. Touch Screen Driver Introduction

5.1 GT911 Introduction

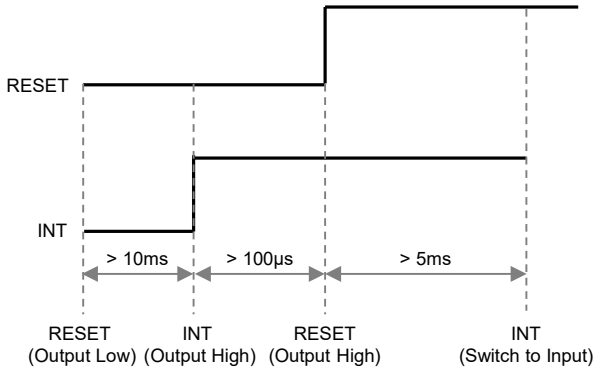
The capacitive touch driver IC for the ESK32-A4A32 module is the GT911. This chip supports simultaneous recognition of the positions, movement tack and touch areas of five touch points, and can read touch information for the corresponding points according to the host's requirements. It communicates via an I²C interface and operates as a slave device, with all communication initiated by the host. It is recommended that the I²C communication rate be configured to 400kbps or below. Furthermore, the I²C device address of the GT911 comprises a 7-bit device address plus one read/write control bit. Its communication address as a slave device is configurable, with two address options provided.

7-Bit Address	8-Bit Write Address	8-Bit Read Address
0x5D	0xBA	0xBB
0x14	0x28	0x29

The host configures the slave address during power-on initialization by controlling the RESET and INT states. The specific method and timing diagrams are as follows:

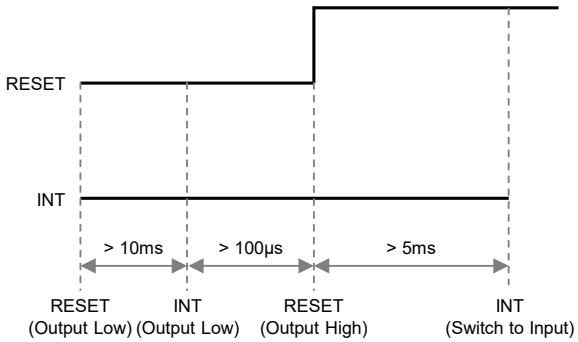
(1) Configure the RESET and INT pins to output mode.

A. To set the I²C address to 0x28 (write) / 0x29 (read), pull both RESET and INT low for a delay exceeding 10ms. Subsequently, pull INT high with a 100μs delay, and then pull RESET high with a 5ms delay.



GT911 I²C Address Configuration 1

B. To set the I²C address to 0xBA (write) / 0xBB (read), pull both RESET and INT low for a delay of 10ms, continue to delay by 100μs after the system stabilizes, and then pull RESET high with a 5ms delay.

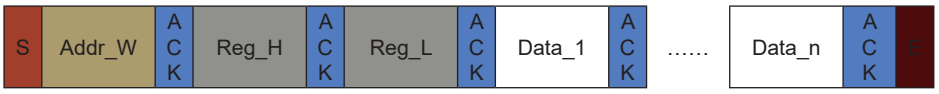


GT911 I²C Address Configuration 2

(2) Set INT to floating input mode and maintain RESET as a high output to complete the device address configuration.

5.2 Data Transmission

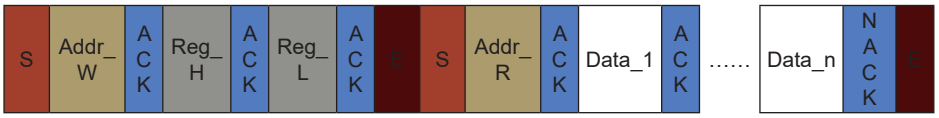
(1) The write operation procedure performed by the host on the GT911 is as follows (using device addresses 0xBA/0xBB as an example):



The host first generates a start signal, then transmits the device address information and the read/write bit information '0' (indicating a write operation): 0xBA. Upon receiving the acknowledgement, the host sends the 16-bit register address, followed by the 8-bit data content to be written into the register.

The register address pointer of the GT911 automatically increments by one following a write operation. Consequently, when the host requires writing to registers at consecutive addresses, it may write data sequentially in a single write operation. Upon completion of the write operation, the host transmits a stop signal to terminate the current write operation.

(2) The read operation procedure performed by the host on the GT911 is as follows (using device addresses 0xBA/0xBB as an example):



The host first generates a start signal, then transmits the device address information and the read/write bit information "0" (indicating a write operation): 0xBA. Upon receiving an acknowledgement, the host sends a 16-bit address information to set the address of the first register to be read. After receiving another acknowledgement, the host re-transmits the start signal and sends the read operation address: 0xBB. Upon receiving a further acknowledgement, the host begins reading data.

The stop signal (the first E signal as shown in the above diagram) after setting read operation register address is optional. However, the start signal to restart I²C communication has to be resent.

The GT911 also supports continuous read operations, with continuous data reading enabled by default. After receiving each byte of data, the host must send an acknowledgment signal to indicate successful reception. Upon receiving the final required byte of data, the host sends a "non-acknowledgment signal (NACK)" followed by a stop signal to terminate communication.

5.3 Obtain Touch Point Information

When a touch event occurs, the GT911's INT pin outputs a rising or falling edge signal, and the corresponding registers respond to the change simultaneously. For a detailed description of the GT911's registers, refer to its datasheet. This document only introduces several commonly used registers.

(1) Status Register

Register Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x814E	Buffer status	Large detect	Reserved		Number of valid touch points			

For this register, Bit7 indicates the status of the touch point data buffer. When the GT911 detects a valid touch point, it automatically sets Bit7 to 1. Therefore, Bit7 can be used to determine whether the screen is being touched. Bit6 indicates the large area detection status, 1 means there is a large-area touch on the touch pad. Bits 3~0 represent the number of valid touch points, ranging from 0 to 5, allowing users to determine how many touch points are present on the screen.

Note that after reading this register, if Bit7 is set high, it must be reset to zero by writing a value of 0. Otherwise, the GT911 will not output the next data.

(2) Touch Point Information Registers

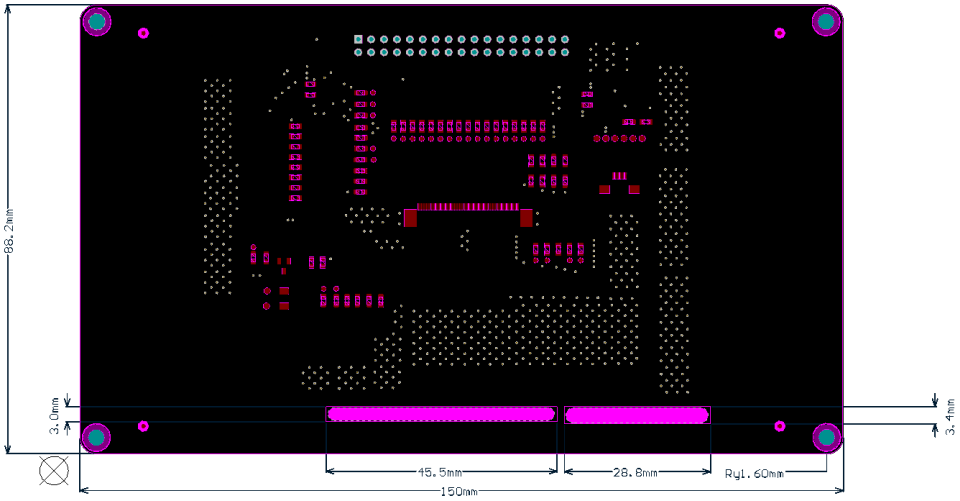
The touch point information registers are divided into five groups (GT911 supports up to 5 touch points), with each group containing six registers. The following table illustrates the registers assigned for five touch points.

Register Address	Description	Register Address	Description
0x8150	Point 1 X coordinate (low byte)	0x8151	Point 1 X coordinate (high byte)
0x8152	Point 1 Y coordinate (low byte)	0x8153	Point 1 Y coordinate (high byte)
0x8154	Point 1 size (low byte)	0x8155	Point 1 size (high byte)
0x8158	Point 2 X coordinate (low byte)	0x8159	Point 2 X coordinate (high byte)
0x815A	Point 2 Y coordinate (low byte)	0x815B	Point 2 Y coordinate (high byte)
0x815C	Point 2 size (low byte)	0x815D	Point 2 size (high byte)
0x8160	Point 3 X coordinate (low byte)	0x8161	Point 3 X coordinate (high byte)
0x8162	Point 3 Y coordinate (low byte)	0x8163	Point 3 Y coordinate (high byte)
0x8164	Point 3 size (low byte)	0x8165	Point 3 size (high byte)
0x8168	Point 4 X coordinate (low byte)	0x8169	Point 4 X coordinate (high byte)
0x816A	Point 4 Y coordinate (low byte)	0x816B	Point 4 Y coordinate (high byte)
0x816C	Point 4 size (low byte)	0x816D	Point 4 size (high byte)
0x8170	Point 5 X coordinate (low byte)	0x8171	Point 5 X coordinate (high byte)
0x8172	Point 5 Y coordinate (low byte)	0x8173	Point 5 Y coordinate (high byte)
0x8174	Point 5 size (low byte)	0x8175	Point 5 size (high byte)

Since the six register addresses corresponding to a single touch point are consecutive, and the GT911 supports auto-incrementing register addresses, programming can begin at the address of the first register for the touch point. By sequentially reading six bytes of data, the X and Y coordinates along with the size information for the touch point can be obtained.

6. Structure Dimensions

The structure dimension of the ESK32-A4A32 module is shown in the figure below. The module size is 150mm×88.2mm, and the mounting hole radius is 1.6mm. The center of the mounting holes is positioned 3.2mm from the edge of the board frame.



ESK32-A432 PCBA Structure Diagram