



VK3816 Datasheet

16-channel touch / 3 groups of slider I2C output

Rev.1.1

Intellectual Property Statement:

Shenzhen Vinka Microelectronics Co., Ltd. (hereinafter referred to as “the Company”) owns legally registered intellectual property rights in both domestic and international jurisdictions. Any unauthorized use of the Company’s products or patented technologies by individuals or organizations is strictly prohibited.

The Company reserves the right to take legal action against any infringement, and to seek full compensation for damages or unlawful gains.

The Company’s name and trademarks are legally protected and may not be used or imitated without explicit written permission. No implied or express license shall be granted under any circumstances.

1 General Description

This software provides customers with a simple slider button application solution.

Customers can set and read slider button and independent button touch data by using the IIC communication format.

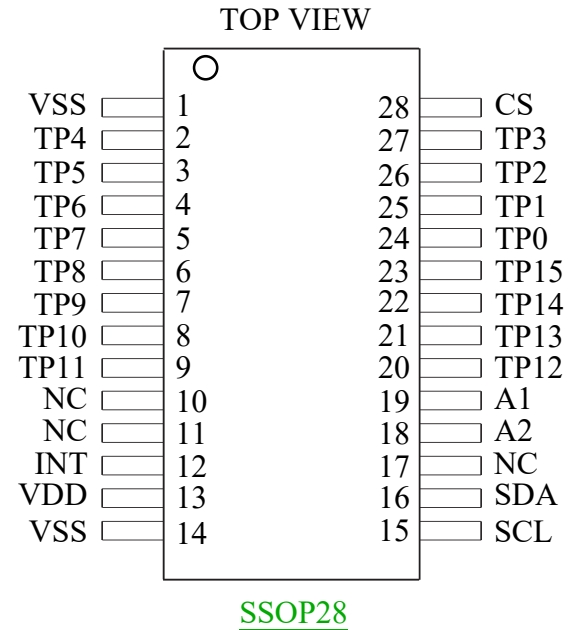
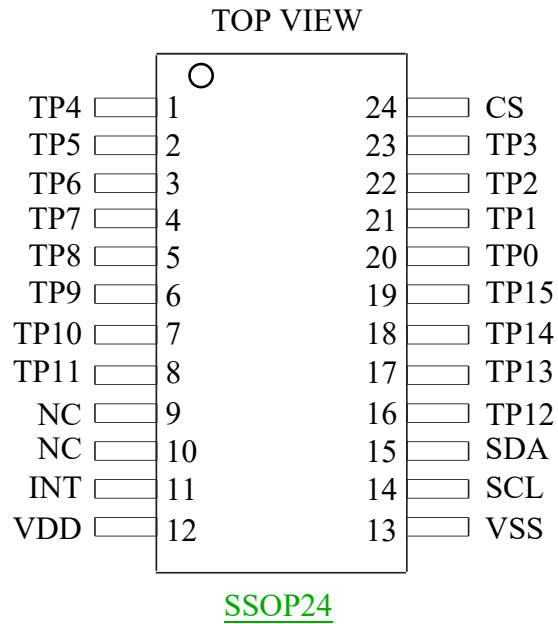
2 Key Features

- The app uses 16 Touch pads to allow customers to plan according to their design needs. The 16 Touch Pads can be planned as slider keys or independent keys, and the sliders can be planned as up to 3 groups of sliders at most. When no slider button settings are made, all 16 Keys can be used as independent buttons.
- Changes there are two ways to set parameters, users can cooperate with USB PCLink Board to adjust the slider button touch sensitivity parameters such as, then using IIC written instruction to modify the setting parameters.
- If Multiple is selected, all keys pressed will be printed. If Single is selected, only the first key pressed will be printed. Only the next key will be printed when the key is released.
- Power saving mode is designed, suitable for use in need long time standby applications such as remote control.
- There are 4 groups of Slave address, can by an external switch.
- Available Packages:
SSOP24 (150mil) (8.65mm × 3.90mm PP=0.635mm)
SSOP28(150mil) (9.9mm × 3.9mm PP=0.635mm)

3 Application field

- Large and small household appliances
- Access control and surveillance equipment
- Consumer electronics

4 Package Pinout Information(SSOP24/SSOP28)



For more information: Page 34-35

Ps: SSOP24 pin package cannot choose IIC device address, can only use the default values.
 CS and CAPN are measurement capacitor pins, and the capacitance size is about 10nF~39nF.
 TP0 to TP15 are measurement pads for touch keys, while the VK3816 can detect up to 16 keys at most. REFL value bolt lock the input for the environment, empty for high level, will cease lower environmental values to update.

A1, A2 is the Slave ID choice of IIC, behind will set mode.

SDA is the data output/input pin of IIC.

SCL is the frequency input pin of IIC.

4.1 VK3816/SSOP28/SSOP24 Pin Description

No.		Pin name	I/O	Function
SSOP28	SSOP24			
13	12	VDD	VDD	Positive power supply
14	13	VSS	VSS	Negative power supply
28	24	CS	I	Touch sensor input
1	13	VSS	VSS	Negative power supply
17	—	NC	—	NC
12	11	INT	I	Touch state output and open-drain output require an external pull-up resistor
11-10	10-9	NC	—	NC
15	14	SCL	I	I2C serial clock pin, open drain output needs external pull-up resistor
16	15	SDA	I/O	I2C serial data input/output pins, open-drain output requires an external pull-up resistor
18-19	—	A1-A2	I	I2C slave address selection pin
24-27 2-5	20-23 1-4	TP0-TP7	I	Touch input
6-9 20-23	5-8 16-19	TP8-TP15	I	Touch input

5 Electrical Characteristics

5.1 Maximum Rated Values

Parameter	Symbol	Conditions	Value	Unit
Operating temperature	Top	—	-40~+85	°C
Storage temperature	TSTG	—	-50~+125	°C
Power supply voltage	VDD	Ta=25°C	VSS-0.3~VSS+5.5	V
Input voltage	VIN	Ta=25°C	VSS-0.3~VDD+0.3	V
Chip anti-static strength HBM	ESD	—	>5	KV

Note: VSS represents system grounding.

5.2 DC/AC Characteristics: (Test conditions: room temperature = 25°C)

Parameter	Symbol	Test conditions	Min.	Typ.	Max.	Unit
Operating voltage	VDD		2.5	-	5.5	V
System oscillation frequency	F	VDD=5V	-	4M	-	HZ
Working current	IOP	Work, VDD = 3V, output without load	-	1.1	-	mA
	IOFF	Standby, VDD=3V output no load	5.3	6.8	10.0	uA

6 Function Description

Touch Button Introduction:

Touch the keys is that, when using measuring human close to the conductor capacitance change, converted to a way of value judgment. All the touch keys in this application have a Threshold setting parameter, which is used to adjust the sensitivity of the touch keys.

The Threshold is adjusted according to the pressing depth of the key. The smaller the value, the more sensitive it is, but the more susceptible it is to noise interference. The user needs to cooperate with the USB PC Link Board operation and adjust it according to the actual pressing and reading data.

Slider Introduction:

The principle of slider button is to use the measure on PCB LAYOUT touch press depth, a method to parse the press position. Advantages of parsing out the available at least keys address most keys. Slider graphics mainly divided into the ring with straight two application, the following figure:

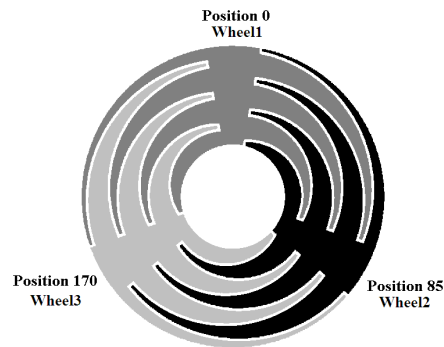


Figure 1. Circular Design



Figure 2. Bar Design

Its principle is to use the pressure values obtained from the Touch Pad of change, then using the differential method to calculate the relative address. Therefore, we need a minimum of 3 keys to calculate the difference between the deepest button and the left and right buttons.

Design Suggestions on key and center distance should be less than 30 mm. The distance between tooth and tooth is about 0.4 mm (pictured), general with 3 ~ 4 teeth design is preferred.

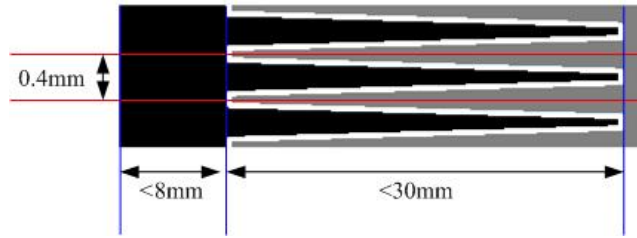


Figure 3. Key Points of Layout Design

Slider button can be set up 3 ~ 16 key, when set to 3 key using the TP0 ~ TP2, TP3 ~ TP15 can planning for general keys; If the slider key is set to 4, TP0~TP3 is used, and TP4~TP15 can be planned as a general key.

As shown in the table:

Slide 1	Disable	3Key	3Key	4Key	Disable	Disable	16Key
Slide 2	Disable	Disable	3Key	4Key	8Key	Disable	Disable
Slide 3	Disable	Disable	Disable	4Key	8Key	4Key	Disable
TP0	Key 1	Slide 1_1	Slide 1_1	Slide 1_1	Slide 2_1	Slide 3_1	Slide 1_1
TP1	Key 2	Slide 1_2	Slide 1_2	Slide 1_2	Slide 2_2	Slide 3_2	Slide 1_2
TP2	Key 3	Slide 1_3	Slide 1_3	Slide 1_3	Slide 2_3	Slide 3_3	Slide 1_3
TP3	Key 4	Key 1	Slide 2_1	Slide 1_4	Slide 2_4	Slide 3_4	Slide 1_4
TP4	Key 5	Key 2	Slide 2_2	Slide 2_1	Slide 2_5	Key 1	Slide 1_5
TP5	Key 6	Key 3	Slide 2_3	Slide 2_2	Slide 2_6	Key 2	Slide 1_6
TP6	Key 7	Key 4	Key 1	Slide 2_3	Slide 2_7	Key 3	Slide 1_7
TP7	Key 8	Key 5	Key 2	Slide 2_4	Slide 2_8	Key 4	Slide 1_8
TP8	Key 9	Key 6	Key 3	Slide 3_1	Slide 3_1	Key 5	Slide 1_9
TP9	Key 10	Key 7	Key 4	Slide 3_2	Slide 3_2	Key 6	Slide 1_10
TP10	Key 11	Key 8	Key 5	Slide 3_3	Slide 3_3	Key 7	Slide 1_11
TP11	Key 12	Key 9	Key 6	Slide 3_4	Slide 3_4	Key 8	Slide 1_12
TP12	Key 13	Key 10	Key 7	Key 1	Slide 3_5	Key 9	Slide 1_13
TP13	Key 14	Key 11	Key 8	Key 2	Slide 3_6	Key 10	Slide 1_14
TP14	Key 15	Key 12	Key 9	Key 3	Slide 3_7	Key 11	Slide 1_15
TP15	Key 16	Key 13	Key 10	Key 4	Slide 3_8	Key 12	Slide 1_16

Definitions of wheel pads and key pads

Slider button need in accordance with the number order, to the right position, prohibit any change order.

7 IIC Agreement

IC uses IIC data transfer protocol, two wire bus SCL, SDA to read and write data. The INT pin is used to notify the Master of any changes in the key status.

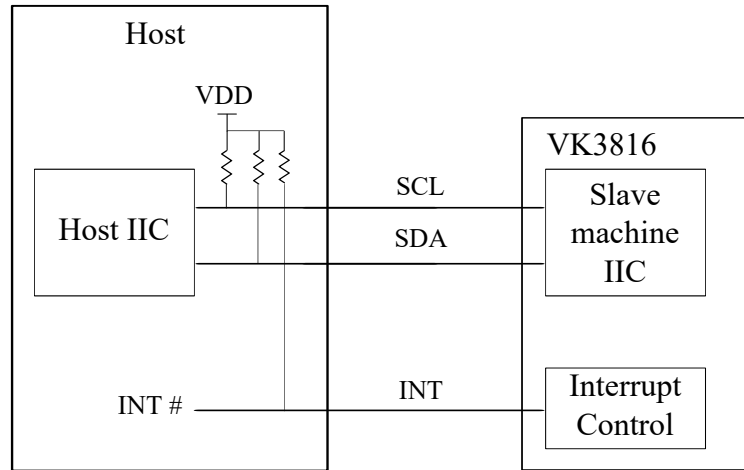


Figure 4. The host and VK3816 are connected via IIC.

INT is high when there is no key state change, and when there is a key state change, the INT pin will pull the bottom 100ms.

If the Slave receives the Slave address, it clears back to high.

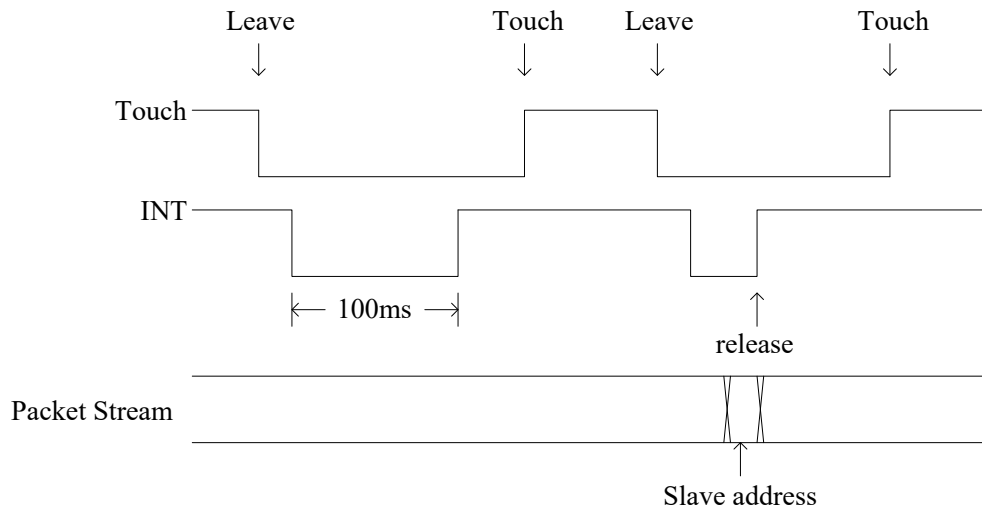


Figure 5. Description of INT pin

7.1 Characteristics of the Switch

Symbol	Description	Min.	Max.	Unit
FSCL	SCL clock frequency	0	100	KHz
THDSTA	Keep the time (repeat) star state. After this period, the first clock pulse	4.0	-	us
TLOW	SCL clock low period	4.7	-	us
THIGH	High SCL clock cycle	4.0	-	us
TSUSTA	Set a time of repeated start condition	4.7	-	us
THDDAT	Data retention time	0	-	us
TSUDAT	Data setting time	250	-	ns
TSUSTO	Set a time to stop state	4.0	-	us
TBUF	In spare time between the stop and start conditions	4.7	-	us
TSPI	The pulse width of pulse is input filter	0	50	ns
TSPT	From processor time	10	75	us

Ac characteristics of IIC SDA and SCL pins for VDD

7.2 Time Waveform

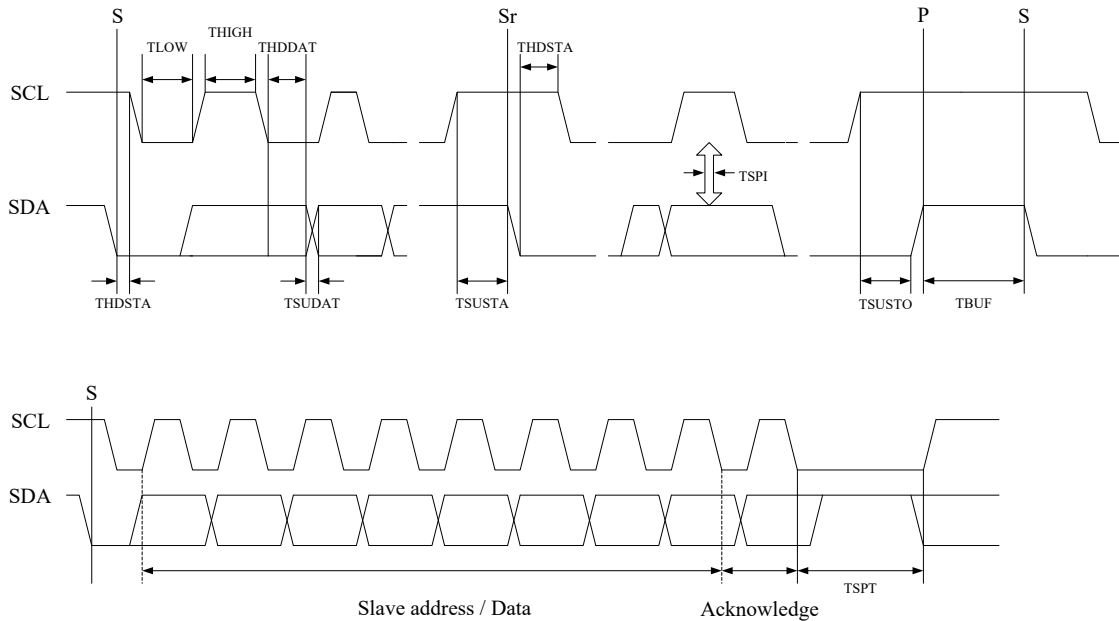


Figure 6. Timing definitions for fast/standard modes on the IIC

7.3 Data Packet Flow

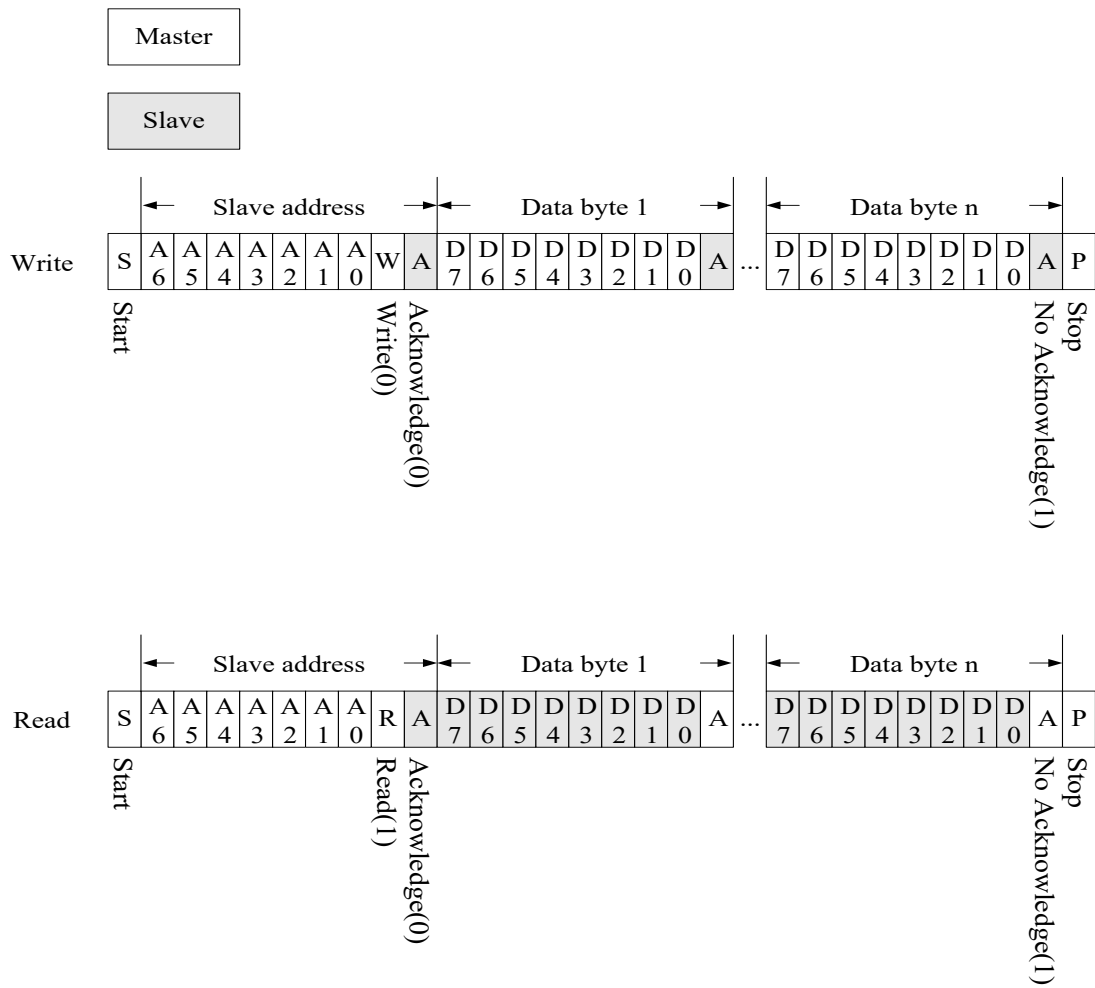


Figure 7. Writing/Reading Bytes via I2C

7.4 Slave device address

It can be switched through pins A1 and A2. As shown in the table below:

A1	A2	Slave address (A6-A0)	Write (A6-A0,R)	Read (A6-A0,R)
0	0	50H	A0H	A1H
0	1	51H	A2H	A3H
1	0	52H	A4H	A5H
1	1	53H	A6H	A7H

* A1, A2 air connection Slave address is 53H

7.5 Data flow

There are two working modes for software design: one is the PC Link mode, and the other is the slider application mode. The PC Link mode requires the cooperation of the USB PCLink Board to read the touch count value, which is used to set the appropriate pressing threshold (Threshold). The slider application mode can set system parameters and read the key status as well as the output of the Wheel. When the first 7 bits of the written data are 0, the system is set to the PC Link mode; if they are 1, it is set to the slider mode.

In the slider application mode, the written data is a data stream of 3 to 4 Data Bytes. After a data stream of data is written completely, the system will overwrite the data and reset the system. If the writing is interrupted and rewritten, the previous data will be discarded.

After each group of 3 to 4 Data Bytes is written, if you want to write the next set of settings again, you need to send the Stop signal to end the current data transmission and then rewrite the next set of settings.

8 Application Mode

Write data

8.1 Set Command

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=0	KOM	AA	PSM	DT	ART	
2	Key Num					KAT		
3	Slide 2 Num				Slide 1 Num			
4	Key Off Num				Slide 3 Num			

8.1.1 IICM

IIC data mode selection.

IICM	IIC Mode
0	PC Link mode
1	Slide application mode (Define)

8.1.2 CT

In the Slide application mode, the data to be written is divided into application settings and threshold settings. When CT is 0, it is for writing application settings; when CT is 1, it is for writing threshold settings.

CT	Custom Threshold
0	Setting commands
1	Custom threshold commands

8.1.3 KOM

Key output mode, have more than one key output and single output two modes. This option is the output setting for the normal key, and the slider button is not affected. In the single-key output mode, only the first pressed key will be output. Other keys will be recognized only after the key is released.

KOM	Key Output Mode
0	Multiple
1	Single (Define)

8.1.4 AA

Automatic adjustment, the basic value when no buttons, automatic updates at baseline.

AA	Auto Adjust
0	Disable
1	Enable (Define)

8.1.5 PSM

Power-saving mode, without buttons, enters sleep mode after 4 seconds.

PSM	Power Save Mode
0	Disable
1	Enable (Define)

Static current: 6.8uA @ 3.0V Operating current: 1.1mA @ 3.0V

8.1.6 DT

Dynamic threshold. When the function is enabled, the threshold value of the sliding button will be automatically adjusted according to the press position.

DT	Dynamic Threshold
0	Disable (Define)
1	Enable

8.1.7 ART

Automatically reset the time setting. The timer starts when the button position remains unchanged, and it resets automatically when the time is up.

ART		Auto Reset Time
0	0	Disable
0	1	15 second (Define)
1	0	30 second
1	1	60 second

8.1.8 KAT

Button shake suppression time.

KAT			Key Acknowledge Times
2	1	0	
0	0	0	1 times
0	0	1	2 times
0	1	0	3 times
0	1	1	4 times (Define)
1	0	0	5 times
1	0	1	6 times
1	1	0	7 times
1	1	1	8 times

8.1.9 Key Num

Key count setting: When the slider is set to "Disable", the maximum number of keys for ordinary keys is 16 Keys. When the number of ordinary keys is set to 16 Keys, if there is a slider key setting, the number of ordinary keys will be the maximum key count of 16 minus the number of slider keys.

Key Num					Key Number
4	3	2	1	0	
0	0	0	0	0	Disable
0	0	0	0	1	1 key
0	0	0	1	0	2 keys
0	0	0	1	1	3 keys
0	0	1	0	0	4 keys
0	0	1	0	1	5 keys
0	0	1	1	0	6 keys
0	0	1	1	1	7 keys
0	1	0	0	0	8 keys
0	1	0	0	1	9 keys
0	1	0	1	0	10 keys
0	1	0	1	1	11 keys
0	1	1	0	0	12 keys
0	1	1	0	1	13 keys
0	1	1	1	0	14 keys
0	1	1	1	1	15 keys
1	0	0	0	0	16 keys (Define)

8.1.10 Slide x Num

The number of slider buttons can be set up to a maximum of 16.

Slide x Num				Slide x Number
3	2	1	0	
0	0	0	0	Disable (Define)
0	0	0	1	Disable
0	0	1	0	3 keys Slide
0	0	1	1	4 keys Slide
0	1	0	0	5 keys Slide
0	1	0	1	6 keys Slide
0	1	1	0	7 keys Slide
0	1	1	1	8 keys Slide
1	0	0	0	9 keys Slide
1	0	0	1	10 keys Slide
1	0	1	0	11 keys Slide
1	0	1	1	12 keys Slide
1	1	0	0	13 keys Slide
1	1	0	1	14 keys Slide
1	1	1	0	15 keys Slide
1	1	1	1	16 keys Slide

8.1.11 Key Off Num

Multiple key reset settings, with a maximum of 16 keys.

Judgment is made based on the number of pressed keys for Slide1, Slide2, Slide3, and Normal respectively.

Key Off Num				Key Off Number
3	2	1	0	
0	0	0	0	Disable (Define)
0	0	0	1	2key reset
0	0	1	0	3 keys reset
0	0	1	1	4 keys reset
0	1	0	0	5 keys reset
0	1	0	1	6 keys reset
0	1	1	0	7 keys reset
0	1	1	1	8 keys reset
1	0	0	0	9 keys reset
1	0	0	1	10 keys reset
1	0	1	0	11 keys reset
1	0	1	1	12 keys reset
1	1	0	0	13 keys reset
1	1	0	1	14 keys reset
1	1	1	0	15 keys reset
1	1	1	1	16 keys reset

8.2 Custom Threshold Command

The threshold is the level at which the button activation is recognized. It is divided into two types: button threshold and sleep wake-up threshold.

8.2.1 Item

Select the option to switch between different write parameter settings.

Item	Item	Item
0	0	TPx setting
0	1	Sleep setting
1	0	-
1	1	-

8.2.2 TPx Setting

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=0		TP Num			
2	TPx Threshold M				TPx Threshold L			
3					TPx Threshold H			

8.2.2.1 TPx Threshold M

TPx Threshold: Button Acknowledgment Threshold. (Define : 010H)

Pressing the button increases the sensitivity as the threshold value decreases, and vice versa. The preset threshold value is 010H, and the recommended minimum value is 008H. If adjusting to 008H still fails to achieve sufficient sensitivity, it is recommended to increase the CS capacitor. The value of the CS capacitor should be less than 39nF.

8.2.2.2 TP Num

The expected value and threshold setting of the buttons are arranged according to the position of the touch buttons. If the slider button is 3 keys and the regular button is 13 keys, then TP0 - TP2 are slider buttons and TP3 - TP15 are regular buttons.

The button numbers for data writing.

TP NUM				TP Number
3	2	1	0	
0	0	0	0	TP0
0	0	0	1	TP1
0	0	1	0	TP2
0	0	1	1	TP3
0	1	0	0	TP4
0	1	0	1	TP5
0	1	1	0	TP6
0	1	1	1	TP7
1	0	0	0	TP8
1	0	0	1	TP9
1	0	1	0	TP10
1	0	1	1	TP11
1	1	0	0	TP12
1	1	0	1	TP13
1	1	1	0	TP14
1	1	1	1	TP15

8.2.3 Sleep Setting

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		-			
2	TPSLP Threshold M				TPSLP Threshold L			
3					TPSLP Threshold H			

TPSLP Threshold: Power-saving Mode Wakeup Threshold. (Define : 002H)

8.3 Read Data

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	C	WSET				S3T	S2T	S1T
2	Key 8	Key 7	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1
3	Key 16	Key 15	Key 14	Key 13	Key 12	Key 11	Key 10	Key 9
4	S1 Position							
5	S2 Position							
6	S3 Position							

8.3.1 C

System correction flag: When the value is 0, it indicates that system correction is in progress and the key value reading is invalid. When the value is 1, the key value is valid.

C	Calibrate
0	Calibrating
1	Calibrate Finish

8.3.2 WSET

The system writes a flag. When powered on, it is set to 1. After the setting is written, this flag is set to 0.

WSET	Have write setting
0	Have write setting
1	No write setting

8.3.3 SxT

The slider button indicator is 0 when there is no button pressed and 1 when a button is pressed.

SxT	Slide x Touch
0	No touch
1	Touch

8.3.4 K1...K16

Touch key flag, no key is 0, with a key is 1.

K1...K16	Key1...Key16
0	No touch
1	Touch

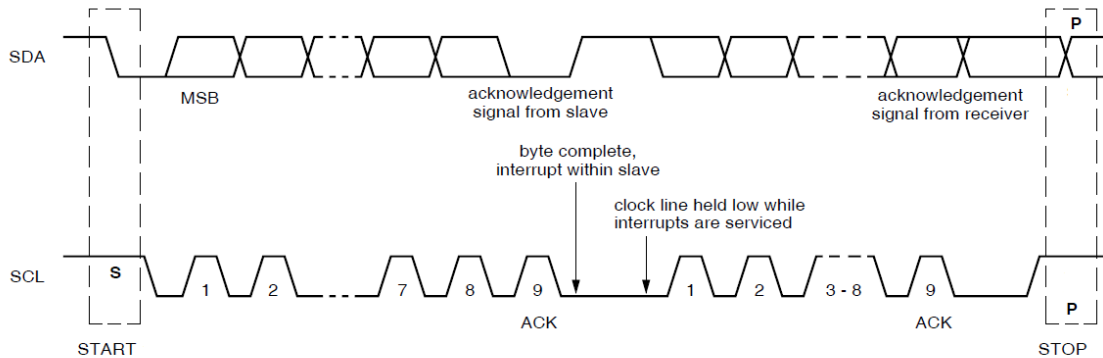
8.3.5 Sx Position

Slider position marker. Preset as 0. Touch the slider to output the press position, and release to reserve the last press position.

Position								Position
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Position 0
0	0	0	0	0	0	0	1	Position 1
0	0	0	0	0	0	1	0	Position 2
x	x	x	x	x	x	x	x	...
1	1	1	1	1	1	0	1	Position 253
1	1	1	1	1	1	1	0	Position 254
1	1	1	1	1	1	1	1	Position 255

9 Special Note

1. The I2C interface of VK3816 has hardware support SCL, which can support 100KHz, but the decoding is processed by software. Therefore, when the 9th SCL of Master is Low, VK3816 will immediately pull the bus of SCL Low, indicating that VK3816 enters the busy state. At the same time, VK3816 will generate interrupt processing I2C decoding, processing needs about 20~100us depending on the processing situation, etc. After processing, it will release SCL, generally the main control SCL control pin is the output of Nmos, need to add a pull resistance. In case the master cannot pull SCL High.



Therefore, when the Master writes the program, it is necessary to pay attention to the action of SCL pulling Low. If it is controlled by the hardware, most of them will support this standard. If the IO foot is controlled by the process, please increase the SCL output High to read back to confirm that it is High, so that the program can continue. C program is as follows:

```
SCL=1;
While(SCL!=1) { };
```

2. If you need to read the key value continuously, it is recommended to pause more than 10ms after reading the key value before reading the next key value. Otherwise it will affect the reaction speed of buttons.

3. If open the sleep mode, then banned continuous read the keys, because every time, while reading the key value will be cleared to sleep time, can lead to system can't sleep.

4. The system into sleep mode, the IIC function will be closed. At this point to the IIC instruction can awaken the system, but will receive no response ACK, needs to wait for the system again after awakening to the read/write command.

5. Steps for adjusting the key threshold:

Step 1. Step1. Select CS capacitor for initial test (see suggested circuit diagram):

First, determine whether the slider function is used in the design. If the slider function is used, it is recommended to use 33nF as the CS charge and discharge capacitor. If it is only used for general keys, it is recommended to use 10nF as the initial test capacitor.

Step 2. Conduct pressing tests for each button:

Using the normal speed to gently touch the button or using a metal rod for the test conditions. If there is a button output before touching the button, it indicates that the sensitivity is too high and the threshold needs to be increased. If touching the button does not result in a button output or only a button output occurs upon pressing harder, it indicates that the sensitivity is too low and the threshold needs to be decreased. The slider button has a serrated design, so the sensitivity varies at different positions. Therefore, it is recommended to conduct the sensitivity test at the midpoint between the two serrated buttons to avoid poor sliding effect.

Step 3. Test button response speed:

When determining the button sensitivity, if you feel that the button is "not sensitive enough", you need to further determine whether it is the response speed of the button that is not fast enough or the sensitivity of the button is not sufficient. The method is to touch for a period of time (about 1 second) and check if there is a button output. If there is no button output, it indicates that the button is not sensitive enough, and you need to re-adjust according to Step 2. If there is a button output, it indicates that the response speed of the button is not fast enough, and proceed to the next step.

Step 4. Adjust button response speed:

The button debouncing time (KAT) is preset to 4. If the button response speed is not fast enough, the value can be lowered to 3. If lowering it to 3 still does not result in a fast response speed, it is recommended to reduce the CS capacitor. After selecting an appropriate CS capacitor, return to Step 2 to re-adjust the sensitivity. It should be noted that choosing a smaller CS capacitor will also reduce the precision of the slider button.

10 Model procedure

```
/*
Project Name: Sample program for master control of VK3816 through IIC
Project Objective: 1. Simulate the operation of the IIC master controller to write setting
parameters to VK3816 using software.
2. Through software simulation, the IIC master controller reads the key status of VK3816.

MainControllerMCU:AT89C51
Date & Version:2018/01/06 v1.0
*/ //-----
#include<reg51.h>
#include<intrins.h>
#define uint unsigned int
#define uchar unsigned char
#define address_W 0xa6 //Slave's address and write flag
#define address_R 0xa7 //The slave's address and read flag

sbit SINT=P0^0; // IIC interface between master and slave
sbit SDA=P0^1; // IIC interface between master and slave
sbit SCL=P0^2; // IIC interface between master and slave

uchar Write_Buffer[4]; // Write data cache of the main control end
uchar Read_Buffer[6]; // Read data cache of the main control end
//-----
// Function Name: void delay(uint x)
// Function Purpose: Program delay
// Function Input: x
// Function Output: None
// Intermediate Variables: i
j //-----
void delay(uint x)
{
    uint i,j;
    for(i=x;i>0;i--)
        for(j=0x40;j>0;j--);
} //----- //
Function Name: void sendStart()
```

```
// Function Functionality: The initialization bit of IIC
// Function Input: None
// Function Output: None
// Intermediate Variables: None
//----- void
sendStart() //Start bit
{
    SDA=1; /* Send the initial condition data signal */
    SCL=1;
    while(SCL!=1) {}
    SDA=0; /* Send the starting signal */
    _nop_();
    SCL=0; /* At this position, just output the SCL to 0 and wait for 4US. */
}
//-----
// Function Name: void sendStop()
// Function Purpose: The stop bit for IIC
// Function Input: None
// Function Output: None
// Intermediate Variables: None
//-----
void sendStop() //Stop Position
{
    SCL=0;
    SDA=0; /*Send the data signal indicating the end condition*/
    _nop_();
    SCL=1;
    while(SCL!=1) { };
    _nop_();
    SDA=1;
}
//-----
// Function Name: bit readACK()
// Function Purpose: Read the acknowledge flag bit of IIC
// Function Input: None
// Output: The ACK from the IIC returns 1 for no acknowledge, 0 for acknowledge
// Intermediate Variables: None
//-----
```

```
bit readACK() //Read the response signal
{
    SCL=0;
    SDA=1; /*This is to release the SDA bus, and the slave sends a low level reply*/
    _nop_();
    SCL=1;
    _nop_();
    if(SDA)
        return 1; //no ACK
    else
        return 0; //ACK
}
//-----
// Function Name: void sendACK()
// Function function: The master sends a reply signal
// Function Input: None
// Function Output: None
// Intermediate Variables: None
//-----
void sendACK() //Output response signal
{
    SCL=0;
    SDA=0;
    _nop_();
    SCL=1;
}
//-----
// Function Name: void sendNOACK()
// Function function: The master sends no reply signal
// Function Input: None
// Function Output: None
// Intermediate Variables: None
//-----
void sendNOACK() //Output of no response signal
{
    SCL=0;
    SDA=1;
    _nop_();
}
```

```

    SCL=1;
  }
  //-----
  - // Function Name: void sendByte(uchar dat)
  // Function: Master writes a byte to slave
  // Function Input: dat = The byte to be sent
  // Function Output: None
  // Intermediate Variables: i
  //-----
  -- void sendByte(uchar dat) //Write a byte
  {
    uchar i;
    for(i=0;i<8;i++)
    {
      SCL=0; /*Lock the I2C bus and prepare to send data*/
      if(dat&0x80)
        SDA=1;
      else
        SDA=0;
      _nop_(); /*If resistors need to be connected in series on SDA, SCL, and INT, depending
on the size of the resistors, it is recommended to appropriately extend the time for larger
resistors. Within 100 KHz, this adjustment is sufficient.; */
      _nop_(); SCL=1; /*Here due to the nature of 51 single-chip microcomputer don't
need to do input and output Settings, but if it is other SCM need to pull on the IO mouth to
enter the Settings, read high, SCL to output is high. The first clock falling edge after reading
and writing the ACK will seize the SCL foot for data processing, so set the SCL foot to the
input pull-up and wait for the SCL to be released.。 */
      while(SCL!=1) { };
      dat<<=1;
    }
  }
  //-----
  // Function Name: uchar readByte()
  // Function: Master reads a byte from slave
  // Function Input: None
  // Function Output: The byte that has been read
  // Intermediate Variables: i, dat
  //-----
  uchar readByte() //Read a byte

```

```

{
  uchar i, dat=0;
  for(i=0;i<8;i++)
  {
    SCL=0;
    SDA=1;
    _nop_(); /*If you need to connect resistors in series on SDA,SCL, and INT, it is
    recommended to extend the time appropriately if the resistance is larger than
    100KHZ according to the different sizes of resistors; */
    dat<<=1;
    SCL=1; /*Here, because of the characteristics of 51 MCU, it is not necessary to do input and
    output Settings, but if it is other MCU, it needs to change its IO port to input pull-up Settings
    first. After reading high,SCL turns output to high. After reading and writing the ACK, the first
    clock falling edge slave will clamp the SCL pin for data processing. Therefore, set the SCL pin
    as the input pull-up and wait for the SCL to be released.。 */
    while(SCL!=1) { };
    if(SDA==1)
      dat|=0x01;
  }
  return dat;
}
//-----
//Function name: bit writeIIC(uchar addrW, uchar *writeData, uchar length)
// Function Functionality: Master end writes data to slave device
// Function Input: addrW = Slave device address and write flag
//      *writeData = Prepare the first address of the write data
//      length = The length (in bytes) of the written data
//Function output: Return acknowledge status of IIC communication. If it is 1, stop and return.
//If zero, the return after communication
//Intermediate variable: i,
//ACK //-----
bit writeIIC(uchar addrW, uchar *writeData, uchar length)
{
  uchar i;
  bit ACK;
  sendStart();
  sendByte(addrW); //Transmitting address and writing mark
  ACK = readACK();
  if (ACK)
  {

```

```
        sendStop(); // Incorrect address or device not connected. Send stop signal.
        return ACK;
    }

    for(i = 0; i<length; i++)
    {
        sendByte(writeData[i]);
        ACK = readACK();
        if (ACK)
        {
            sendStop(); //No ACK was received, so the stop signal was sent.
            return ACK;
        }
    }
    sendStop(); //Data writing is complete. Sending stop signal.
    return ACK;
}

//-----
//Function name:bit readIIC(uchar addrR, uchar *readData, uchar length)
//Function function: Master end reads data from slave
//Function input: addrR = Slave address and read flag
//                *readData = The first address to hold data before it is read
//                length = The length (in bytes) of the data being read
//Function output: Return acknowledge status of IIC communication. If it is 1, stop and
//                return. If zero, the return after communication
//Intermediate variable: i
ACK //-----
bit readIIC(uchar addrR, uchar *readData, uchar length)
{
    uchar i;
    bit ACK;
    sendStart();
    sendByte(addrR); //Transmitting address and reading marker
    ACK = readACK();
    if (ACK)
    {
        sendStop(); //If the address is incorrect or the device is not connected, send a stop signal
        return ACK;
    }
}
```

```
    }

    for(i = 0; i<length; i++)
    {
        readData[i] = readByte();
        if(i<length-1)
            sendACK();
        else
            sendNOACK(); //Read the last piece of data and send out No ACK
    }
    sendStop(); //Data reading is complete. Sending stop signal.
    return ACK;
}

//-----
//Function name: void setWrite_Buffer_4(uchar byte1, uchar byte2, uchar byte3, uchar byte4)
//Functionality: Write 4 bytes into the write cache register
//Function input:byte1
//            byte2
//            byte3
//            byte4
// Function output: None
// Intermediate variables: None
//-----
void setWrite_Buffer_4(uchar byte1, uchar byte2, uchar byte3, uchar byte4)
{
    Write_Buffer[0] = byte1;
    Write_Buffer[1] = byte2;
    Write_Buffer[2] = byte3;
    Write_Buffer[3] = byte4;
}

//-----
//Function name: void setWrite_Buffer_3(uchar byte1, uchar byte2, uchar byte3)
// Write 3 bytes to the write cache register
// Function input: byte1
//            byte2
//            byte3
// Function output: None
// Intermediate variables: None
```

```
//-----  
void setWrite_Buffer_3(uchar byte1, uchar byte2, uchar byte3)  
{  
    Write_Buffer[0] = byte1;  
    Write_Buffer[1] = byte2;  
    Write_Buffer[2] = byte3;  
}  
void main()  
{  
    bit ACK;  
    SINT = 1;  
    setWrite_Buffer_4(0xB1, 0x23, 0x33, 0x03);  
    ACK = writeIIC(address_W, &Write_Buffer, 4); //MCU Setting  
    setWrite_Buffer_3(0xC0, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP0 Threshold  
    setWrite_Buffer_3(0xC1, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP1 Threshold  
    setWrite_Buffer_3(0xC2, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP2 Threshold  
    setWrite_Buffer_3(0xC3, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP3 Threshold  
    setWrite_Buffer_3(0xC4, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP4 Threshold  
    setWrite_Buffer_3(0xC5, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP5 Threshold  
    setWrite_Buffer_3(0xC6, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP6 Threshold  
    setWrite_Buffer_3(0xC7, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP7 Threshold  
    setWrite_Buffer_3(0xC8, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP8 Threshold  
    setWrite_Buffer_3(0xC9, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP9 Threshold  
    setWrite_Buffer_3(0xCA, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP10 Threshold  
    setWrite_Buffer_3(0xCB, 0x10, 0x00);  
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TP11 Threshold  
    setWrite_Buffer_3(0xCC, 0x10, 0x00);  
}
```

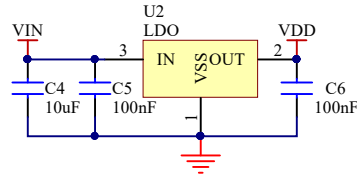
```
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP12 Threshold
setWrite_Buffer_3(0xCD, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP13 Threshold
setWrite_Buffer_3(0xCE, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP14 Threshold
setWrite_Buffer_3(0xCF, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP15 Threshold
setWrite_Buffer_3(0xD0, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Threshold
delay(50);

while(1)
{
    if(!SINT) /*Waiting for the read request. If the power-saving mode is turned off,
there is no need to read the SINT. However, it is recommended to wait for a duration of 30ms
between each read of the key.*/
        ACK = readIIC(address_R, &Read_Buffer, 6); //Read the status of the buttons
    }
}
```

11 Application Circuits

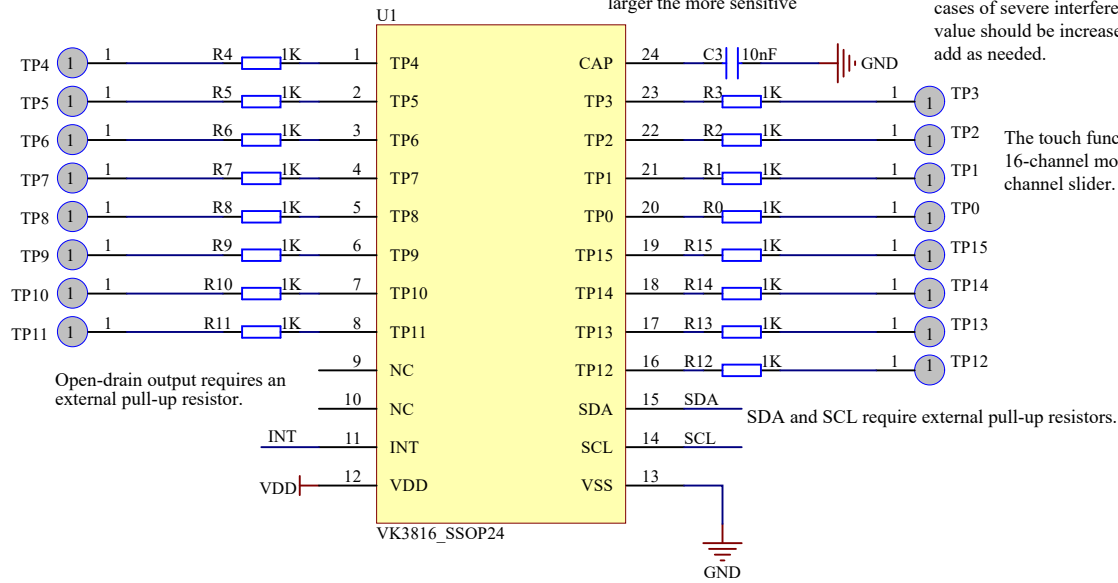
11.1 SSOP24

LDO is recommended for power supply



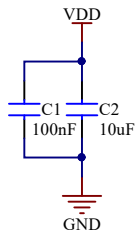
C3 is used to regulate the overall sensitivity generally 1nF~47nF the larger the more sensitive

When touching the resistors connected to the feet, a 0-10K range is usually chosen to enhance the anti-interference capability. In cases of severe interference, the resistance value should be increased. Customers can add as needed.



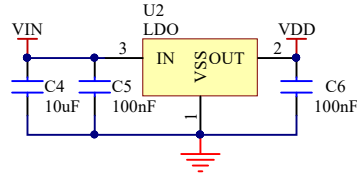
The touch function can be set to 16-channel mode or used as a 3-channel slider.

Power pin and filter capacitor stabilized power supply



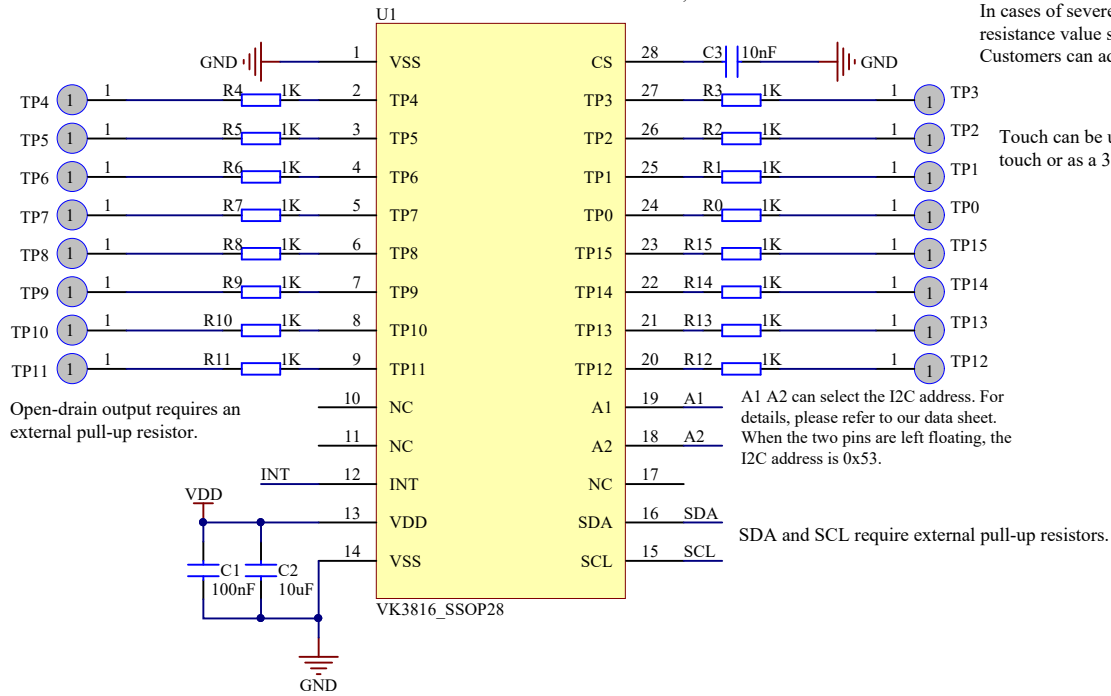
11.2 SSOP28

LDO is recommended for power supply



C3 is used to regulate the overall sensitivity. Generally, the larger the 1nF~47nF, the more sensitive it is

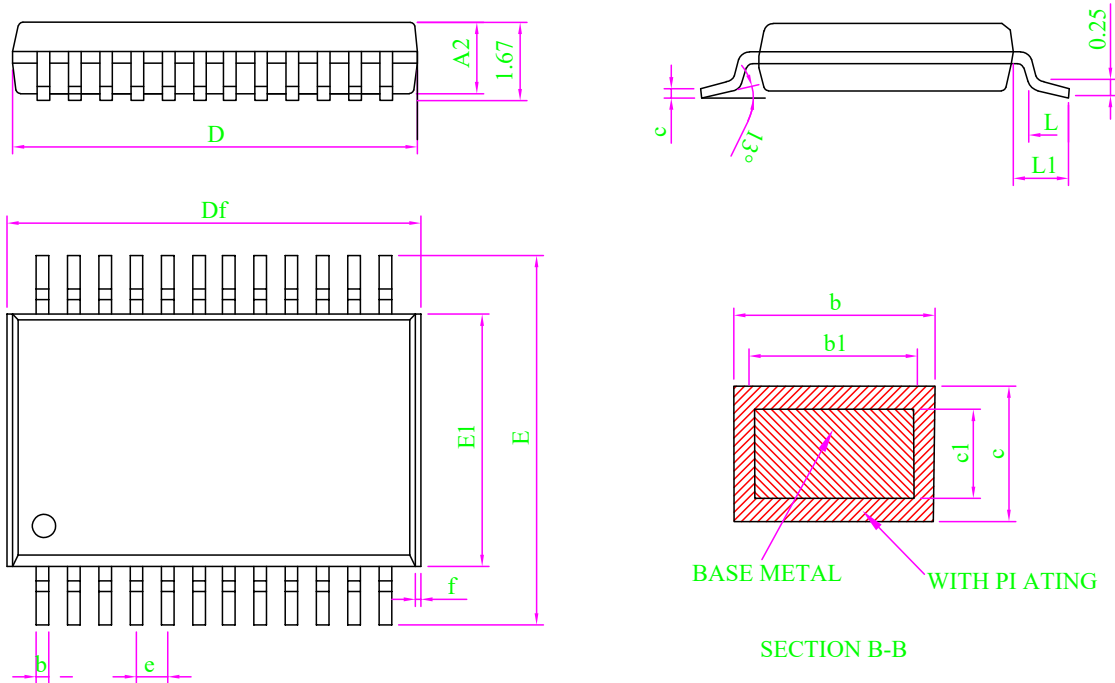
When touching the resistors connected to the feet, a 0-10K range is usually chosen to enhance the anti-interference capability. In cases of severe interference, the resistance value should be increased. Customers can add as needed.



Power pin and filter capacitor stabilized power supply

12 Package Information

12.1 SSOP24 (150mil) (8.65mm × 3.90mm PP=0.635mm)

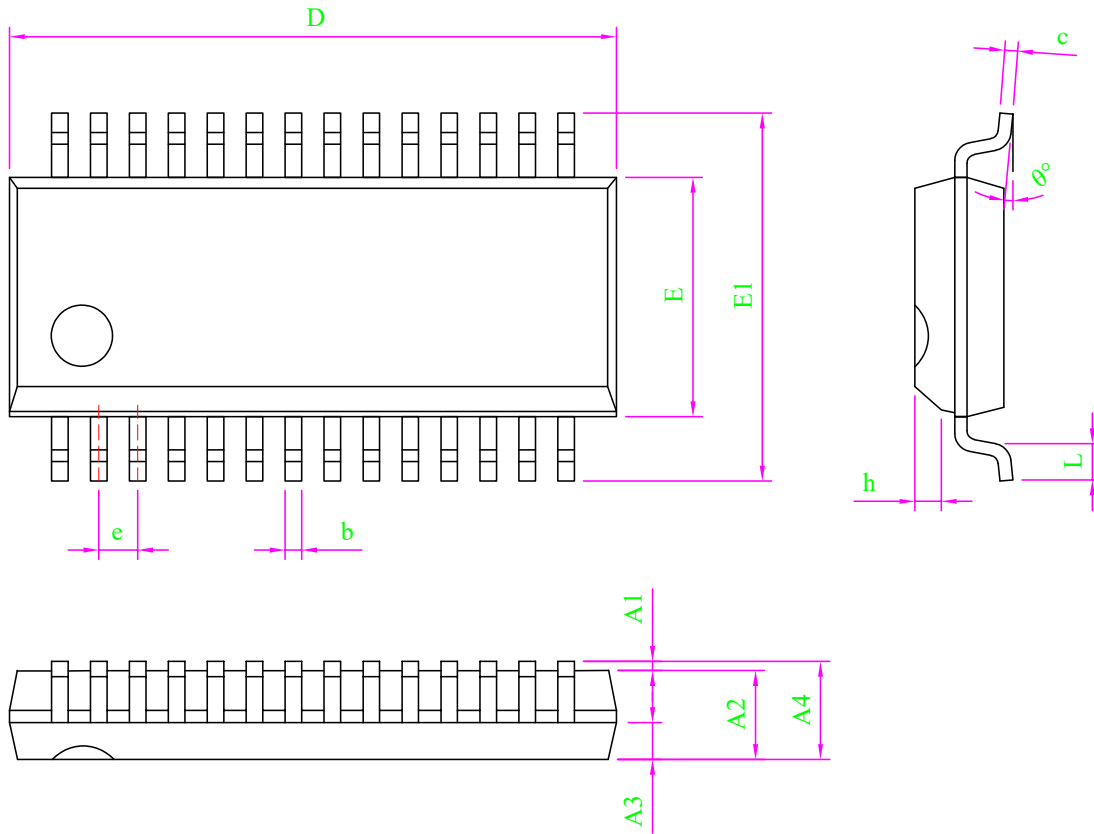


Note:

1. All dimension are in mm.
2. Dim D&E1 does not include plastic flash; Df includes plastic flash(f); Flash: Plastic residual around body edge after de junk/singulation.
3. Dim b does not include dambar protrusion/intrusion.
4. Plating thickness 0.007mm-0.015mm

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.75
A1	0.10	0.15	0.20
A2	1.35	1.45	1.55
b	0.23	-	0.31
b1	0.22	0.25	0.28
c	0.20	-	0.24
c1	0.19	0.20	0.21
D	8.56	8.66	8.76
Df	8.66	-	9.16
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635 BSC		
L	0.51	0.66	0.81
L1	0.95	1.05	1.15
θ	0	-	8°
f	0.05	-	0.20

12.1 SSOP28(150mil) (9.9mm × 3.9mm PP=0.635mm)



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.75
A1	0.05	-	0.23
A2	1.35	1.45	1.55
A3	0.60	0.65	0.70
b	0.23	-	0.31
c	0.19	-	0.25
D	9.80	9.90	10.00
E	3.90	3.90	4.00
E1	5.80	6.00	6.20
e	0.635 BSC		
h	0.30	-	0.50
L	0.60	-	0.80
θ	0	-	8°

13 Disclaimer

Warranty and liability — The information provided in this document is believed to be accurate and reliable. However, Shenzhen Vinka Microelectronics Co., Ltd. (hereinafter “the Company”) makes no warranties, express or implied, as to the completeness or suitability of this information for any specific purpose.

In no event shall the Company be liable for any indirect, incidental, or consequential damages, including but not limited to loss of profits, equipment damage, or system failure, arising out of the use of this product or documentation, regardless of the legal theory under which such liability is asserted.

Right to change — The Company reserves the right to modify any information contained herein without prior notice. The latest version of this document is available at:

<https://www.szvinka.com/>

Applicability — This product is not designed or intended for use in life-critical, medical, or safety systems where failure could result in injury or death. The customer shall assume full responsibility for any such use.

Application — All product application descriptions provided herein are intended for illustrative purposes only. The Company makes no representations or warranties, express or implied, regarding the suitability of any specific application without further testing or modification.

The customer is solely responsible for determining whether the Company’s products are appropriate for their intended applications or end customers.

The customer shall ensure proper design practices, implementation safeguards, and operational validation to minimize risks associated with product use.

The Company shall not be held liable for any defects, losses, costs, or damages arising from weaknesses or failures in the customer’s own products or applications, or from the integration or use of third-party products.

Furthermore, the customer shall conduct all necessary testing and validation for any third-party deployment of the Company’s products to avoid potential misuse or associated damages. The Company assumes no liability in this regard.

Commercial terms of sale — Unless otherwise agreed in writing, sales of this product are subject to the Company’s standard terms and conditions of sale. The Company expressly rejects the applicability of the customer’s general terms and conditions.

Export control — This product may be subject to applicable export control regulations. The customer is solely responsible for compliance with such regulations, including obtaining any necessary export licenses.

14 Revision History

No.	Version	Date	Modify the content	Check
1	1.0	2024-03-22	Initial release	YES
2	1.1	2025-09-08	Change Description	YES

[1] Please refer to the latest version of this document before starting or finalizing any design.

[2] Since the release of this document, the status or availability of this product may have changed. For the most up-to-date information, please visit:

<https://www.szvinka.com/>