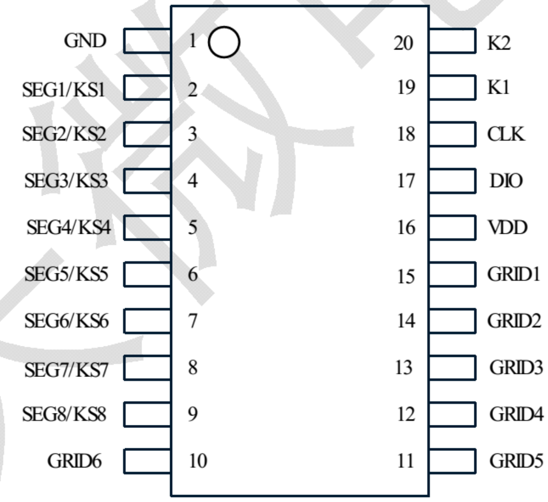
The register stores transferred from an external device via the serial interface to the TM1637 data address 00H-05H bytes of six units, respectively, and SGE and GRID pin chip LED lights are connected to the corresponding distribution in the following figure:

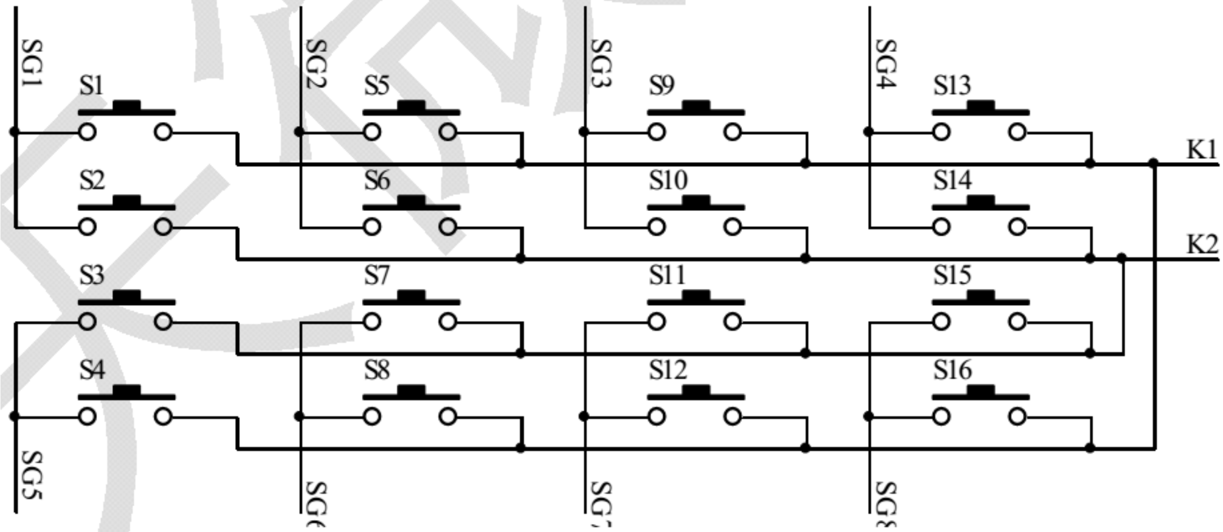


Write LED display data, in accordance with the address from the display from low to high, from low to high data byte operation.

- display mode (8 segment × 6 bits), supporting digital output of the sun
- key scan (8 × 2bit), enhanced anti-jamming circuit identification keys
- brightness adjustment circuit (adjustable duty cycle of 8)
- two-wire serial interface (CLK, DIO)
- Oscillation mode: Built-in RC oscillator (450KHz +5%)
- Built-on reset circuit
- Built-in automatic blanking circuit
- Package: DIP20/SOP20

Symbol	Pin Name	Pin Number	Explanation
DIO	Data input / output Out	17	Serial data input / output, input data changes SLCK low in SCLK high level is transmitted, one byte per chip will be in the first transmission Eight falling clock edges to generate an ACK
CLK	Clock Input	18	The rising edge of the input / output data
K1~K2	Key scan data input	19-20	The data input pin is latched after the end of the display period
SG1~SG8	Output (section)	2-9	Segment output (also used as key scan), N pipe open drain output
GRID6~GRID1	Output (bit)	10-15	Output, P tube open-drain output
VDD	Logic Supply	16	5V±10%
GND	Ground	1	

Key scan matrix is 8 × 2bit, as follows:



When the button is pressed, the key data read as follows:

	SG1	SG2	SG3	SG4	SG5	SG6	SG7	SG8
K1	1110_1111	0110_1111	1010_1111	0010_1111	1100_1111	0100_1111	1000_1111	0000_1111
K2	1111_0111	0111_0111	1011_0111	0011_0111	1101_0111	0101_0111	1001_0111	0001_0111

Note: If no key is pressed, the key data is read: 1111_1111, low front, high in the post. Since the induction cooker and other kitchen appliances applications, due to interference

Strong, in order to improve this problem, TM1637 resolved using negative edge triggered false triggering phenomenon, the so-called "jump key" phenomenon.

The register stores transferred from an external device via the serial interface to the TM1637 data address 00H-05H bytes of six units, respectively, and

SGE and GRID pin chip LED lights are connected to the corresponding distribution in the following figure:

Write LED display data, in accordance with the address from the display from low to high, from low to high data byte operation.

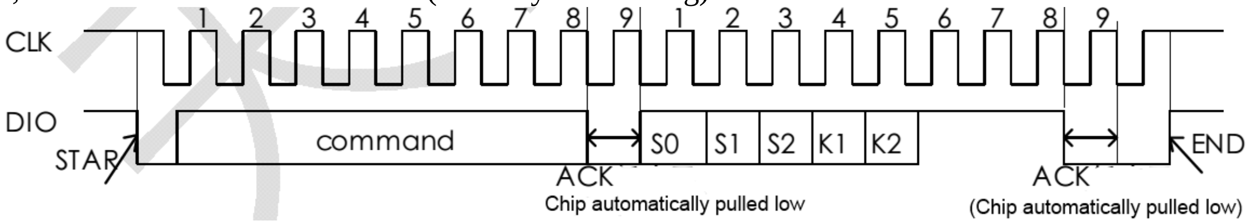
SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	
xxHL (Low four)				xxHU (High four)				
B0	B1	B2	B3	B4	B5	B6	B7	
00HL				00HU				GRID1
01HL				01HU				GRID2
02HL				02HU				GRID3
03HL				03HU				GRID4
04HL				04HU				GRID5
05HL				05HU				GRID6

Interface Description

Microprocessor data via a two-wire bus interface and TM1637 communication, when the input data when CLK is high, the signal on the DIO must remain unchanged; only on the clock

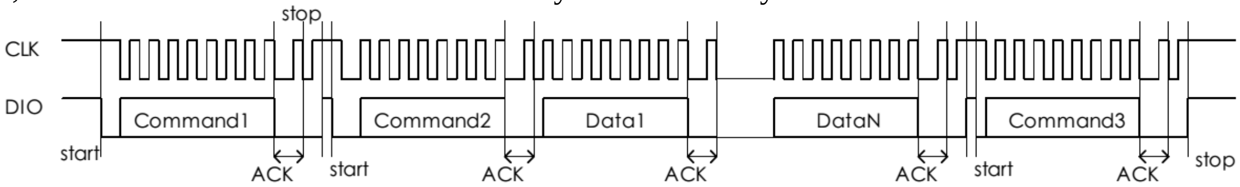
signal CLK is low, the signal can be changed on the DIO. Start condition input data is CLK Is high, DIO from high to low; termination condition is CLK is high, DIO from low to high.
Data transfer TM1637 with the response signal ACK, when transferring data correctly, the falling edge of the eighth clock, the chip will generates an acknowledge signal ACK will DIO pin is pulled low, the release of DIO port line after the end of the ninth clock.

1, data transfer instruction below (read key data timing)



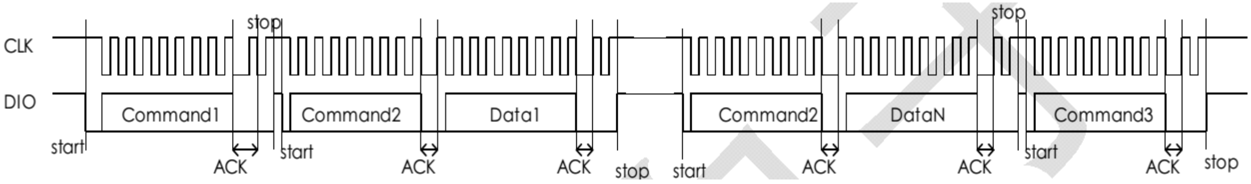
Command: read key commands; S0, S1, S2, K1, K2 constituting the key information is encoded, S0, S1, S2 is encoded SGn, K1, K2 to K1 and K2 key code, read the key, the clock frequency should be less than 250K, first read low, after reading high.

2, SRAM data write address is automatically incremented by 1 mode



Command1: Set Data
Command2: Set Address
Data1 ~ N: display data transmission
Command3: Control Display

3, write SRAM data fixed address mode



Command1: Set Data
Command2: Set Address
Data1 ~ N: display data transmission
Command3: Control Display

Data Directive

Instruction is used to set the display mode and the LED drive status. In the first byte after the falling edge of the CLK input DIO as an instruction. After decoding, to take maximum B7, B6 to distinguish between the two bits different instructions.

B7	B6	Instruction
0	1	Data Command Set
1	0	Display Control Command Set
1	1	Address command set

If you send STOP command in the command or data transmission, the serial communication is initialized, and the instruction or data being transmitted is invalid (before or the data transfer instruction remains valid)

1, the data command set

This command is used to set the data write and read, B1 and B0 are not allowed to set 01 or 11 bits.

B7	B6	B5	B4	B3	B2	B1	B0	Function	Explanation
0	1	Do not care to fill 0				0	0	Data read-write mode settings	Write data to the display register
0	1					1	0		Read key scan data
0	1				0			Address mode is set to increase	Automatic address incrementing
0	1				1				Fixed address
0	1			0				Test mode setting (internal use)	Normal mode
0	1			1					Test Mode

2, set address command set

B7	B6	B5	B4	B3	B2	B1	B0	Show address
1	1	Do not care to fill 0		0	0	0	0	00H
1	1			0	0	0	1	01H
1	1			0	0	1	0	02H
1	1			0	0	1	1	03H
1	1			0	1	0	0	04H
1	1			0	1	0	1	05H

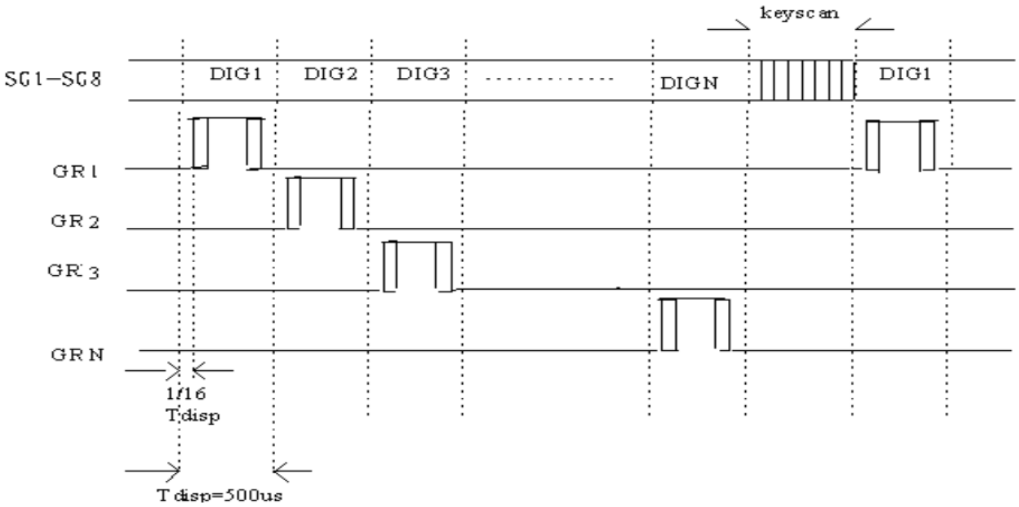
This command is used to set the display address register; If the address is set 0C6H or higher, the data is ignored until a valid address is set;

On power-up, the address defaults to 00H.

3, the display control

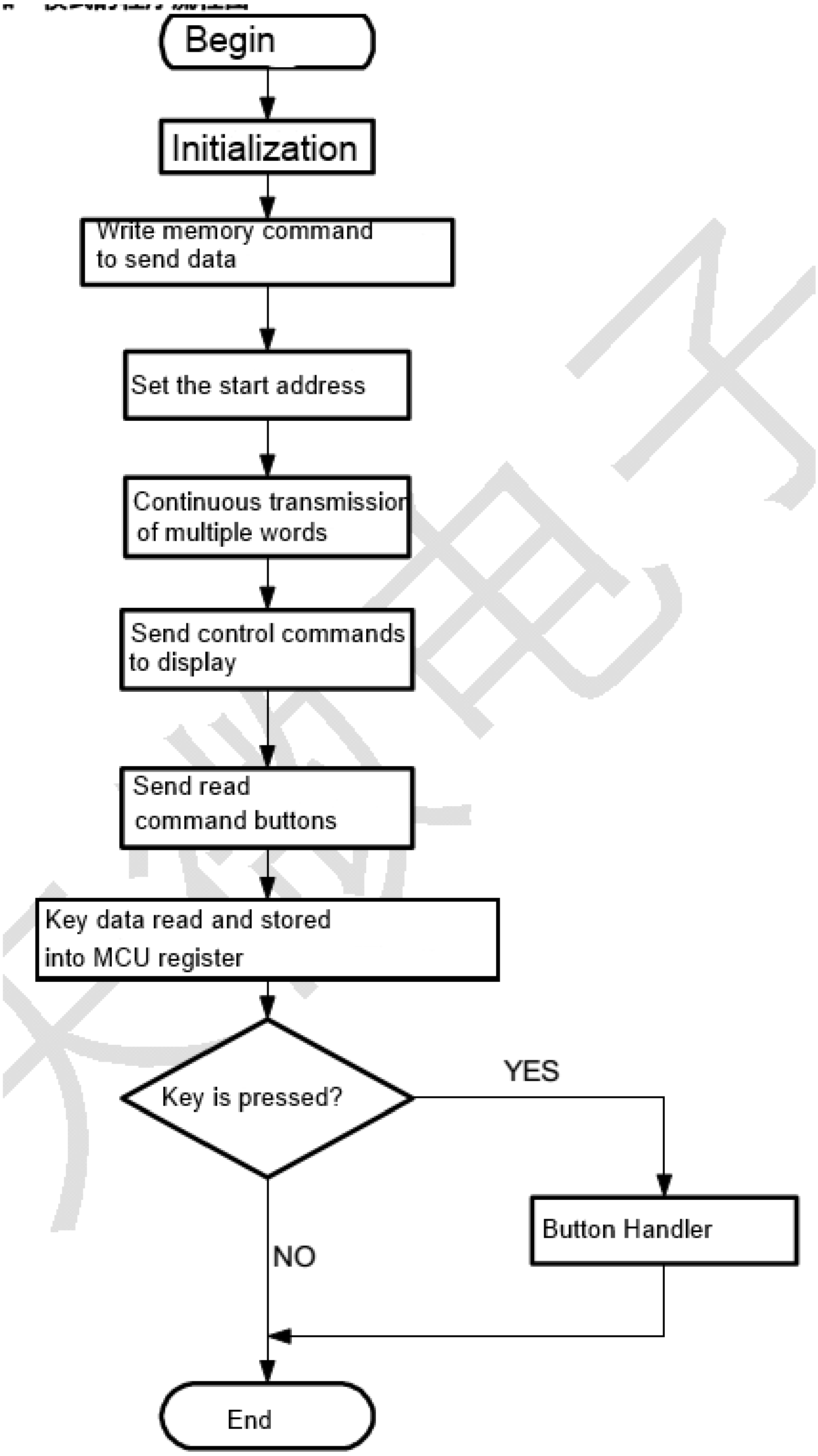
B7	B6	B5	B4	B3	B2	B1	B0	Function	Explanation
1	0	Do not care to fill 0			0	0	0	Extinction number of settings	Set the pulse width of 1/16
1	0				0	0	1		Set the pulse width of 2/16
1	0				0	1	0		Set the pulse width of 4/16
1	0				0	1	1		Set the pulse width of 10/16
1	0				1	0	0		Set the pulse width of 11/16
1	0				1	0	1		Set the pulse width of 12/16
1	0				1	1	0		Set the pulse width of 13/16
1	0				1	1	1		Set the pulse width of 14/16
1	0			0				Display switch settings	Showing Off
1	0			1					Show On

Display and key scan cycle

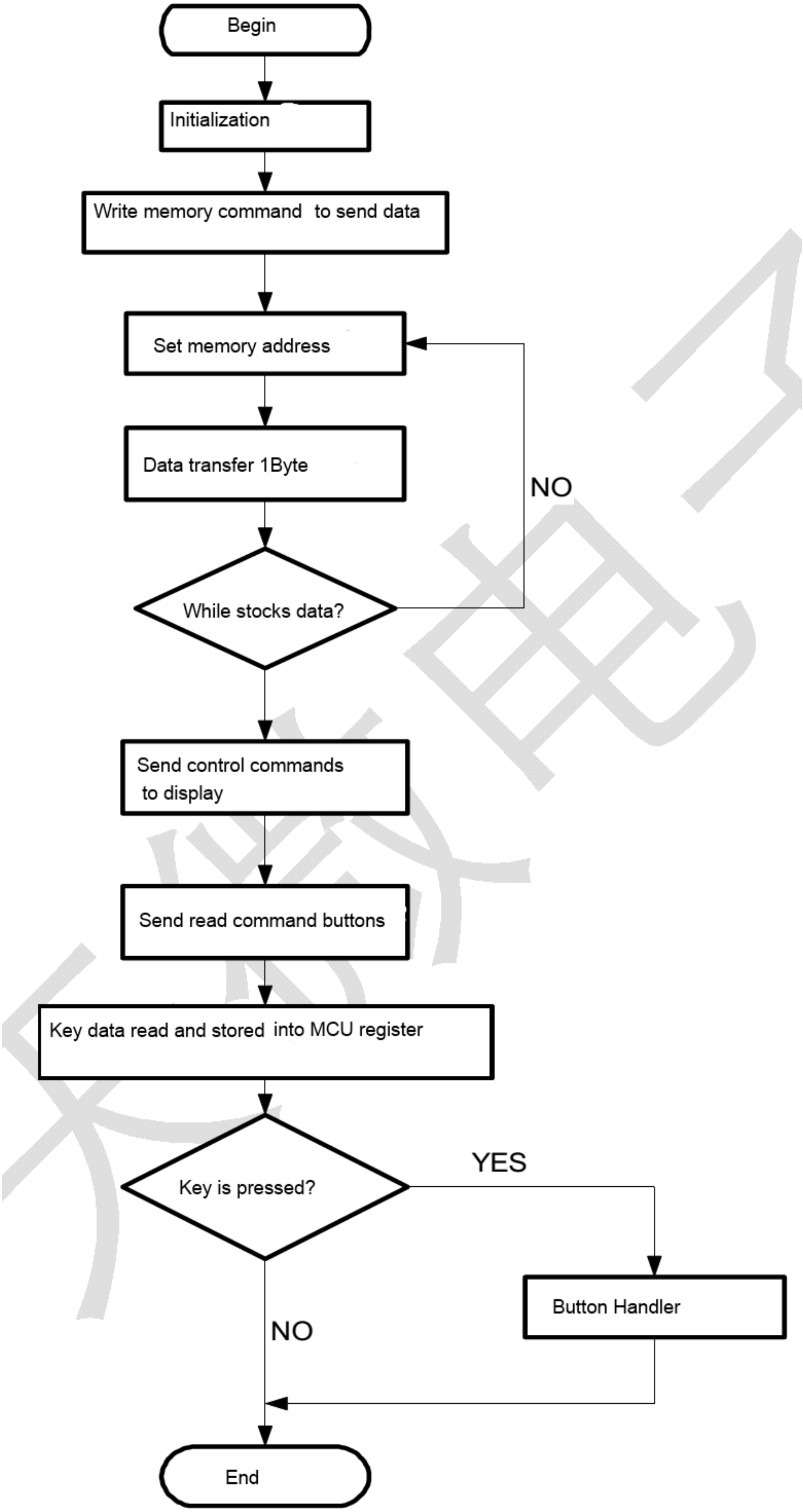


Program flow chart

1, using the address auto mode plus a program flowchart


Begin
Initialization
Write memory command to send data
Set the start address
Continuous transmission of multiple words
Send control commands to display
Send read command buttons
Key data read and stored into MCU register
Key is pressed?
YES
Button Handler
NO
End

2, using a fixed address programming flowcharts


Begin
Initialization
Write memory command to send data
Set memory address
Data transfer 1Byte
While stocks data?
NO
Send control commands to display
Send read command buttons
Key data read and stored into MCU register
Key is pressed?
YES
Button Handler
NO
End

Reference Program

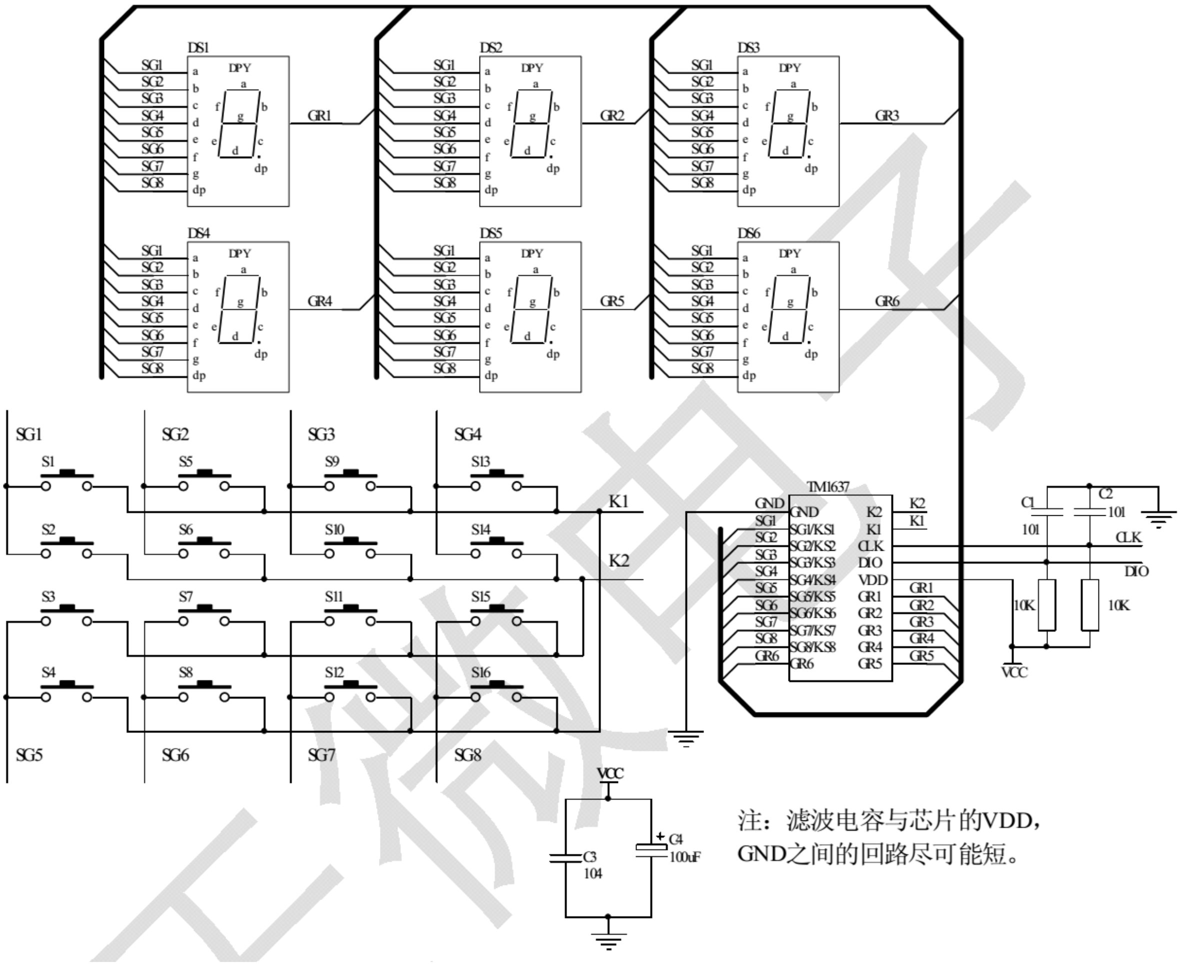
```
/ *
* Copyright: Shenzhen Tian Microelectronics
* File name: TM1637
* Current Version: 1.0
* SCM Model: AT89S52
* Development Environment: Keil uVision3
* Crystal earthquake frequency: 11.0592M
* Program features: the TM1637 all filled with all the display data register
address 0xff, and open the display, and then read the key value.
* /
      # include <reg52.h>
      # include <intrins.h>
      / / Define port
      sbit clk = P1 ^ 2;
      sbit dio = P1 ^ 1;
      / / / =======================================
      void Delay_us (unsigned int i) // n us delay
      {
           for (; i> 0; i--)
           _nop_ ();
      }
      / / / ======================================
      void I2CStart (void) // 1637 start
      {
           clk = 1;
           dio = 1;
           Delay_us (2);
           dio = 0;
      }
      / / / =============================================
      void I2Cask (void) // 1637 Answer
      {
           clk = 0;
           Delay_us (5); // After the falling edge of the eighth clock delay 5us,
ACK signals the beginning of judgment
           while (dio);
           clk = 1;
           Delay_us (2);
           clk = 0;
      }
      / / / ========================================
      void I2CStop (void) // 1637 Stop
      {
           clk = 0;
           Delay_us (2);
           dio = 0;
           Delay_us (2);
           clk = 1;
           Delay_us (2);
           dio = 1;
      }
      / / / =========================================
 void I2CWrByte (unsigned char oneByte) // write a byte
 {
      unsigned char i;
      for (i = 0; i <8; i + +)
      {
            Clk = 0;
            if (oneByte & 0x01) // low front
            {dio = 1;}
            else {dio = 0;}
            Delay_us (3);
            oneByte = oneByte >> 1;
            clk = 1;
            Delay_us (3);
      }
 }
 / / / ----------------------------------------------- -
unsigned char ScanKey (void) // read buttons \
{
      unsigned char rekey, rkey, i;
      I2CStart ();
      I2CWrByte (0x42); // read command buttons
      I2Cask ();
      dio = 1; // read keys before data lines pulled
      for (i = 0; i <8; i + +) // start reading from the low
      {
            Clk = 0;
            rekey = rekey >> 1;

            Delay_us(30);
            clk=1;
            if(dio)
            {
                  rekey=rekey|0x80;
            }
            else
            {
                  rekey=rekey|0x00;
            }
            Delay_us(30);
      }
      I2Cask();
      I2CStop();
      return (rekey);
}

void SmgDisplay(void) // Write display register
{
      unsigned char i;
      I2CStart();
      I2CWrByte(0x40); //40H address is automatically incremented by 1 mode,
44H fixed address mode
      I2Cask();
      I2CStop();
      I2CStart();
      I2CWrByte(0xc0);  // Set the first address
      I2Cask();
      for(i=0;i<6;i++)  // Addresses from Canada, do not always write address
      {
            I2CWrByte(0xff);  // Send data
            I2Cask();
      }
      I2CStop();
      I2CStart();
      I2CWrByte(0x8f); // Open display, maximum brightness
      I2Cask();
      I2CStop();
}
///================================================
void init() // Initialization routine
{
    // Initialization slightly
}
///================================================
void main(void)
{
      unsigned char keydate;
      init() ;  // Initialization
      SmgDisplay(); // Write register and open display
      while(1)
      {
            keydate=Scankey();  // Read key value, the key value is not read
      out for treatment.
      }
}
//===========end===================================
```

Hardware connection diagram

The circuit diagram for the digital connection of positive digital



注：滤波电容与芯片的VDD，GND之间的回路尽可能短。