M031/M032 Emulation Touch Screen Controller

Example Code Introduction for 32-bit NuMicro® Family

Document Information

Application	This sample code uses the M031/M032 series microcontroller (MCU) to read a touch screen signal and analyze the position of the finger touch. This document will describe how to configure the ADC to read the signal, and how to configure the SPI to transmit data to the host.
BSP Version	M031_Series_BSP_CMSIS_V3.05.000
Hardware	NuMaker-M032SE V1.3, NuTFT Kit Board

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. Overview

This sample code can scan the ADC signal to identify the position where the finger touches the screen and uses customized SPI protocol to communicate with the master device through the SPI interface.

1.1 Principle

The four-wire resistive touch panel is composed of two layers of materials with conductive properties (Indium Tin Oxide, ITO). Generally speaking, the upper layer is a conductive film (ITO Film), the lower layer is a conductive glass (ITO Glass), and in the middle of the two layers are Spacer Dots with insulating properties as shown in Figure 1-1. After the touch screen is touched by a finger, an analog voltage drop signal will be generated, and then converted into a digital signal through the controller, thus the correct touch position can be calculated.

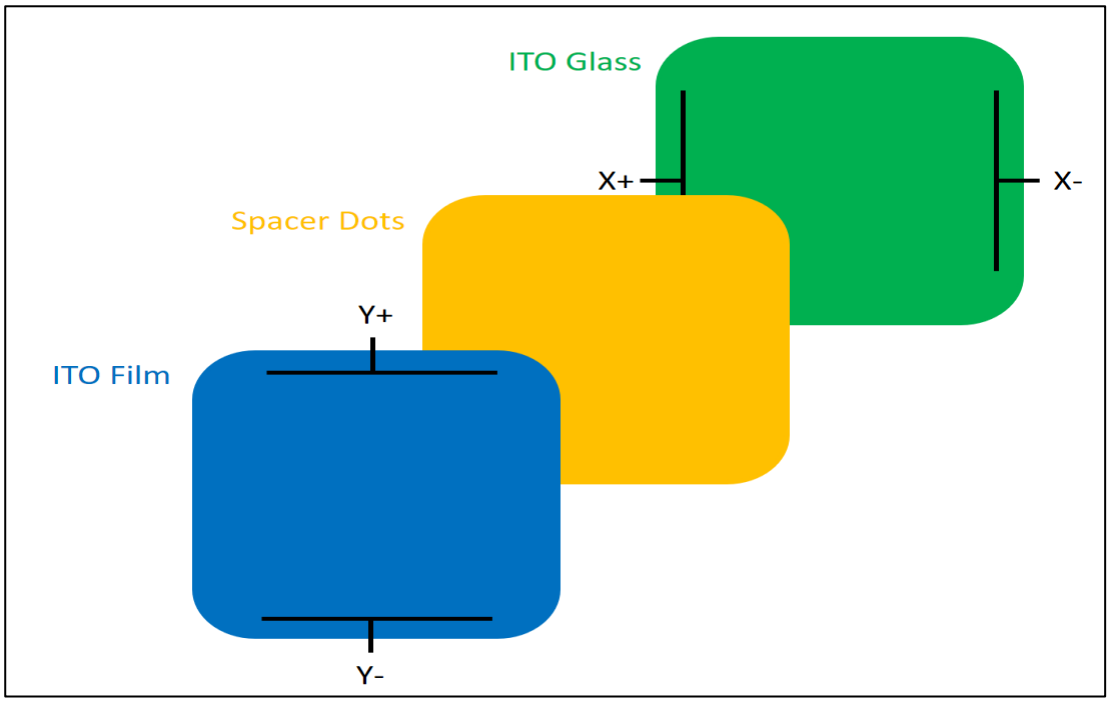

Figure 1-1 4-wire Resistive Touch Panel

1.2 ADC Function

The M031/M032 series MCU uses four GPIO pins to detect the value of x, y screen position and the value of z1 and z2 touched by the finger. Two pins need to be set as VCC and GND respectively, the third pin is on floating state and the last pin uses the ADC function to convert the voltage value of the equivalent resistance of the touch screen. Finally, the x and y positions of the touch are calculated through the MCU.
Figure 1-2 shows the formula for the equivalent resistance and voltage divider of the LCD touch screen. Table 1-1 shows the four GPIO pin settings for the emulated touch screen controller.

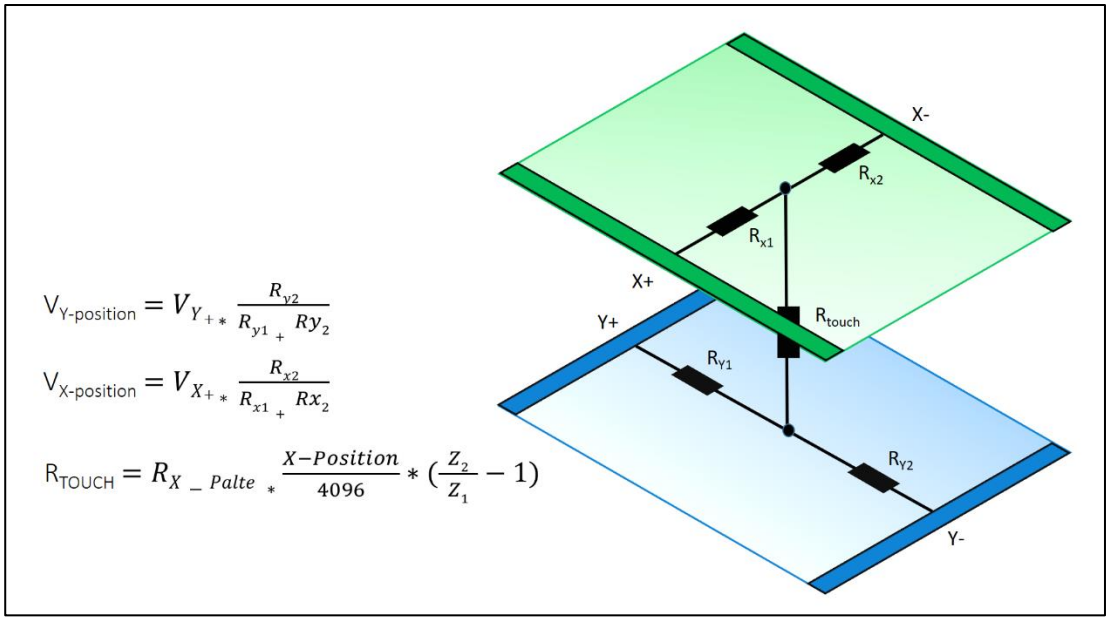

Figure 1-2 Schematic of a 4-wire Touch Screen

	X+	X-	Y+	Y-
X-Coordinate	GND	VCC	Hi-Z	ADC
Y-Coordinate	Hi-Z	ADC	GND	VCC
Z1	GND	ADC	Hi-Z	VCC
Z2	GND	Hi-Z	ADC	VCC

Table 1-1 4-wire Touch Screen Scanning

1.3 SPI Protocol Format

Figure 1-3 shows the customized SPI data protocol. The slave device will read the signal continuously from the touch screen and convert it into digital data. The master device can send commands to the slave device and read the data.

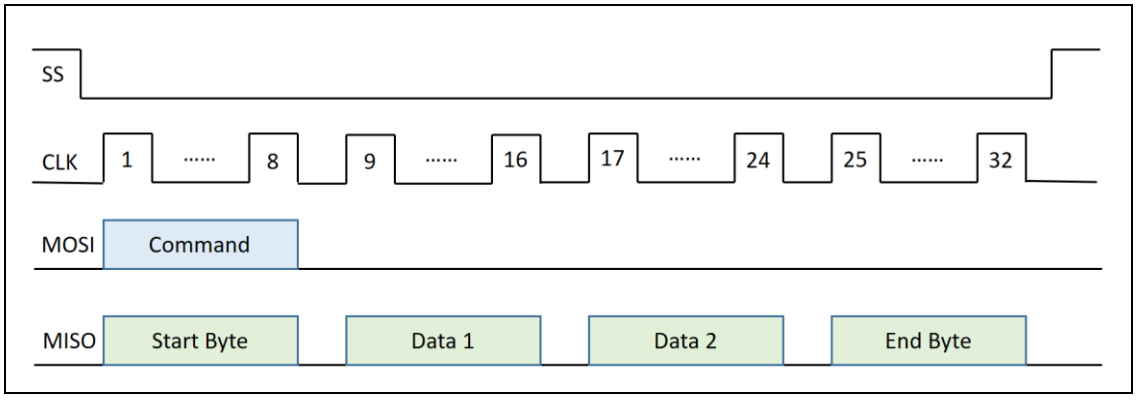

Figure 1-3 SPI protocol

1.4 The Customized Touch Key Protocol

The commands shown in Table 1-2 are used by the master device to request touch screen information on the emulated touch screen controller board.

Command	Start Byte	Respond data1	Respond data2	End Byte
0x01	49	X-Position ADC high byte	X-Position ADC low byte	19
0x02	49	Y-Position ADC high byte	Y-Position ADC low byte	19
0x03	49	Z1-Position ADC high byte	Z1-Position ADC low byte	19
0x04	49	Z2-Position ADC high byte	Z2-Position ADC low byte	19

Table 1-2 Customized Touch Screen Controller Commands

1.5 Demo Results

The demonstration results are shown in Figure 1-4. After the finger touches the touch screen on the slave device, the master device will display the touch data information.

Figure 1-4 Demo Results

2. Code Description

In the main code, always run the Get_X_value, Get_Y_value, Get_Z1_value, Get_Z2_value functions and the delay function. Users can learn how to read the touch screen data through these functions.

```
/* ------------- */
/* Main function */
/* ------------- */
int main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();
    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();
    /* Lock protected registers */
    SYS_LockReg();

    /* Configure UART0: 115200, 8-bit word, no parity bit, 1 stop bit. */
    UART_Open(UART0, 115200);

    /* Init SPI */
    SPI_Init();

    /* Init Priority */
    NVIC_SetPriority(SPI0_IRQn, 0); // Priority high
    NVIC_SetPriority(ADC_IRQn, 2);

    printf("SPI slave test ...\n\n");

    while (1)
    {
        X_Position = Get_X_value();       /* Get X value of Touch LCD */
        Y_Position = Get_Y_value();       /* Get Y value of Touch LCD */
        Z1_Position = Get_Z1_value();     /* Get Z1 value of Touch LCD */
        Z2_Position = Get_Z2_value();     /* Get Z2 value of Touch LCD */

        /* Check the finger is touching or not */
        Touch_or_not(X_Position, Y_Position, Z1_Position, Z2_Position);
        delay_ms(50);       /* Delay */
    }
}
```

3. Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M031_Series_BSP_CMSIS_V3.05.000
- IDE version
  - Keil uVersion 5.36

3.2 Hardware Requirements

- Circuit components
  - NuMaker-M032SE V1.3 (Master device)
  - NuMaker-M032SE V1.3 (Slave device)
  - NuTFT Kit Board (R-type Touch Screen)
- Pin Connect (as shown in Figure 3-1)
  - The XR, XL, YU, YD pins of the NuTFT Kit Board are connected to the GPIO pins of the NuMaker-M032SE (Slave device) to read the touch screen data.
  - The SPI0 of the NuMaker-M032SE (Master device) are connected to the SPI0 of the NuMaker-M032SE (Slave device), and connect to the Ready Pin (PA0) of the master and slave. The Ready Pin indicates that the slave has finished reading the touch screen data, and can start to transfer data.
  - Connect the UART0 to the PC to show the execution results of this example code.

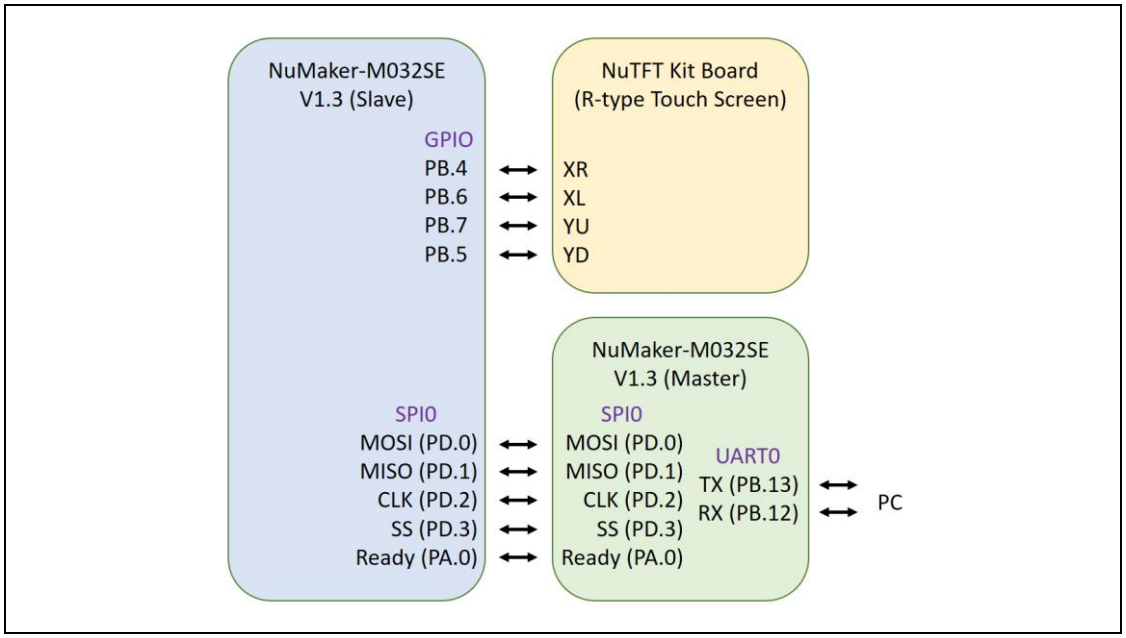

Figure 3-1 Pin Connect

4. Directory Information

The directory structure is shown below.

EC_M031_Emulation_Touch_Screen_Controller_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	Source file of example code

Figure 4-1 Directory Structure

5. Example Code Execution

1. Open Keil uVision5 integrated development environment.
2. Browse the sample code folder *SampleCode\ExampleCode\Emulation_Touch_Screen\Keil* and double-click *Emulation_Touch_Screen.uvprojx.*
3. Enter Keil compile mode.
   - Build
   - Download the program to NuMaker-M032SE V1.3 (Slave)
4. Open HyperTerminal, which is set as Figure 5-1.

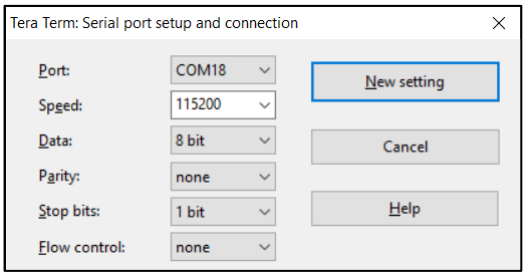

Figure 5-1 HyperTerminal Setting

5. Press the reset button, and the HyperTerminal will display as shown in Figure 5-2.

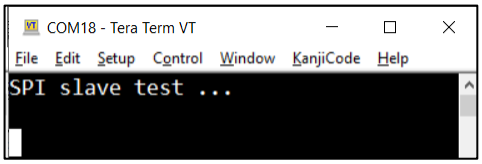

Figure 5-2 SPI Information

6. Reopen Keil uVision5 integrated development environment.
7. Browse the sample code folder *SampleCode\ExampleCode\Emulation_SPI_Master\Keil* and double-click *Emulation_SPI_Master.uvprojx*.
8. Enter Keil compile mode.
   - Build
   - Download the program to NuMaker-M032SE V1.3 (Master)

9. Open HyperTerminal, which is set as Figure 5-3.

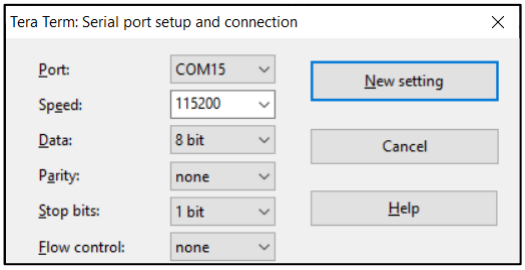

Figure 5-3 HyperTerminal Setting

10. Press the reset button, and the HyperTerminal will display as shown in Figure 5-4.

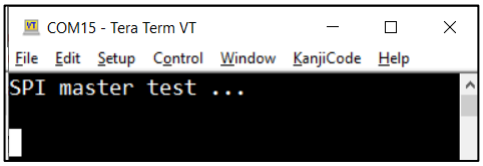

Figure 5-4 SPI Information

11. After touching the screen with your finger, HyperTerminal will display as shown in Figure 5-5.

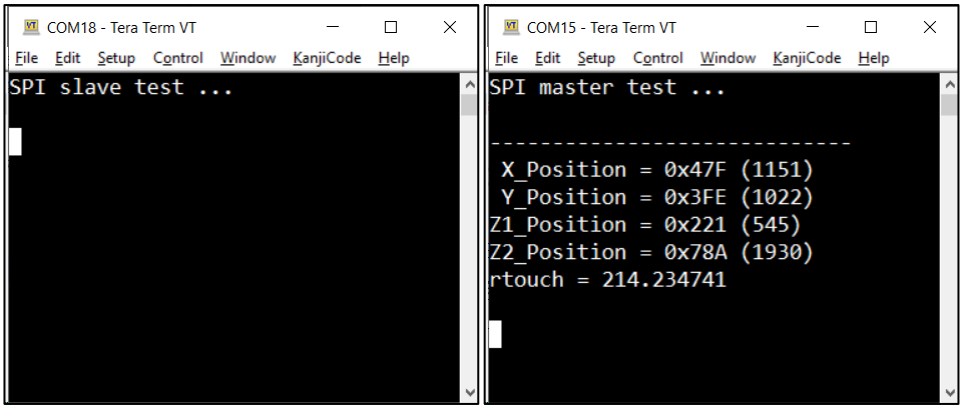

Figure 5-5 Display Results

6. Revision History

Date	Revision	Description
2022.08.24	1.00	Initial version.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.