Application Note

AN_447

FT4222H Communication with Multi SPI Slaves

Version 1.0

Issue Date: 27-07-2023

This document shows how to communicate with multiple SPI slaves when utilizing an FT4222H.

Future Technology Devices International Limited (FTDI)
Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
Web Site: http://ftdichip.com

Table of Contents

1 Introduction

The FT4222H is a High/Full Speed USB2.0-to-Quad SPI/I2C device controller, it contains SPI/ I2C configurable interfaces. The SPI interface can be configured in master mode with single, dual, or quad bits data width transfer or in slave mode with single bit data width transfer.

FT4222H supports 4 operation modes to allow various I2C/SPI devices to be connected to USB bus. The attachable device configuration for each mode is listed below:

- Mode 0 (2 USB interfaces):
  - 1 SPI master, SPI slave, I 2C master, or I 2C slave device
  - 1 GPIO device
- Mode 1 (4 USB interfaces):
  - SPI master connects to 3 SPI slave devices
  - 1 GPIO device
- Mode 2 (4 USB interfaces):
  - SPI master connects to 4 SPI slave devices
- Mode 3 (1 USB interface):
  - 1 SPI master, SPI slave, I 2C master, or I 2C slave device

This document will show when FT4222H is working in mode 2 or mode 3, how the SPI master can communicate with multiple SPI slave devices.

2 Getting Started

A LibFT4222 application usually starts with FT_CreateDeviceInfoList and FT_GetDeviceInfoList much the same as a traditional D2XX application would. Under different chip modes, FT_CreateDeviceInfoList will report a different number of interfaces as shown in the table below.

Mode	Number of Interfaces	Device Function
0	2	a. The first interface: it can be one of SPI master, SPI slave, I 2C master, or I 2C slave device. b. The 2nd interface: GPIO device.
1	4	a. The first 3 interfaces: SPI master connects to 3 SPI slaves. b. The 4th interface: GPIO device.
2	4	a. SPI master connects to 4 SPI slaves, see Figure 2.1.
3	1	a. It can be one of SPI master, SPI slave, I 2C master, or I 2C slave device.

Table 2.1 Chip Mode and Device Functions

After opening the device with FT_Open or FT_OpenEx, developers need to initialize the FT4222H device as either SPI master, SPI slave, I 2C master, or I 2C slave. Different types of devices require different configurations.

Figure 2.1 shows the hardware connection when the FT4222H is configured in mode 2.

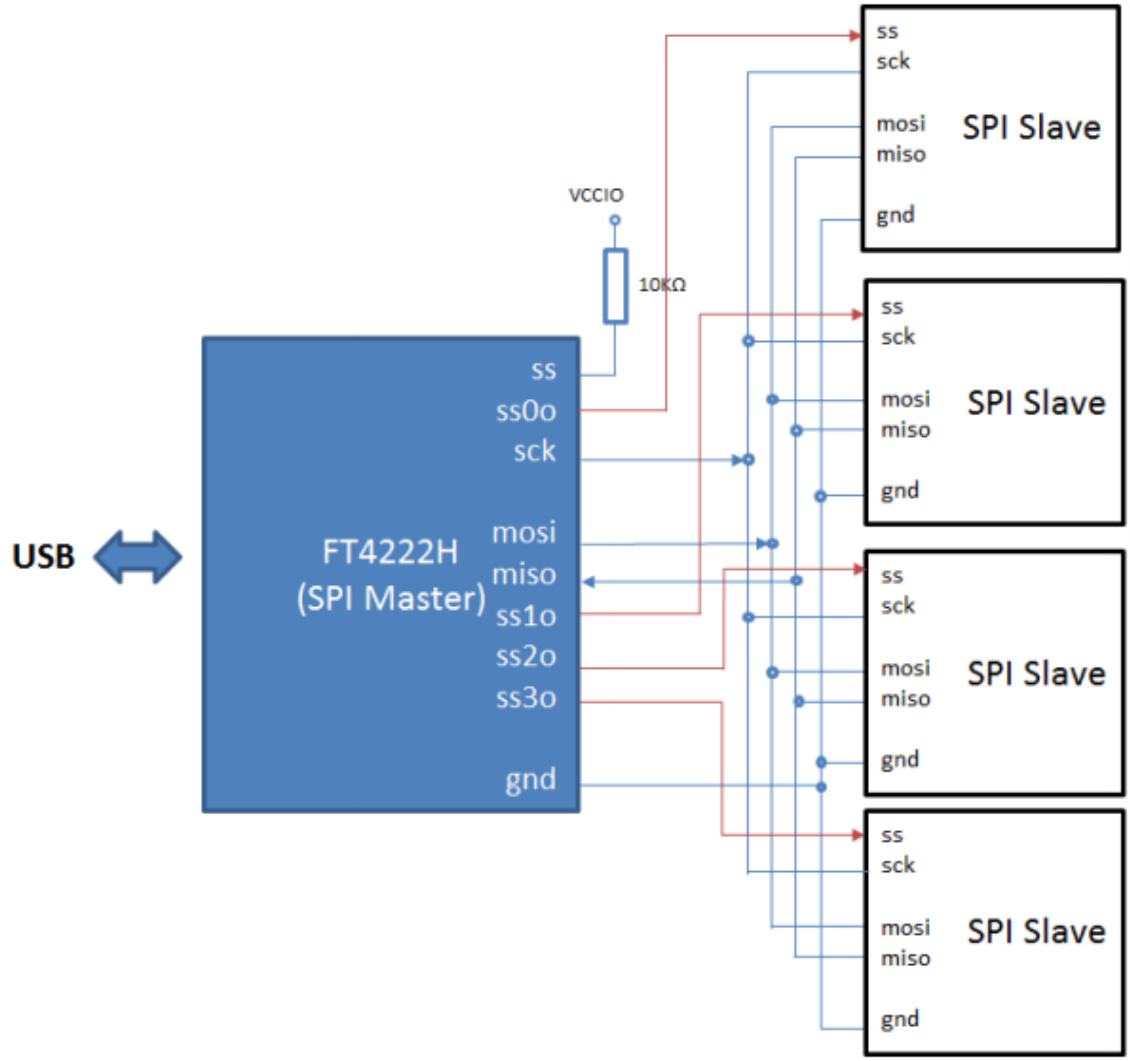

Figure 2.1 Mode 2: FT4222H as an SPI Master

3 Example

LibFT4222 must be used to develop functions, which provide high-level APIs to facilitate user application development. At the time of writing support for Windows and Linux OS has been published. Android support uses a different package also available from the FTDI website.

3.1 Block Diagram

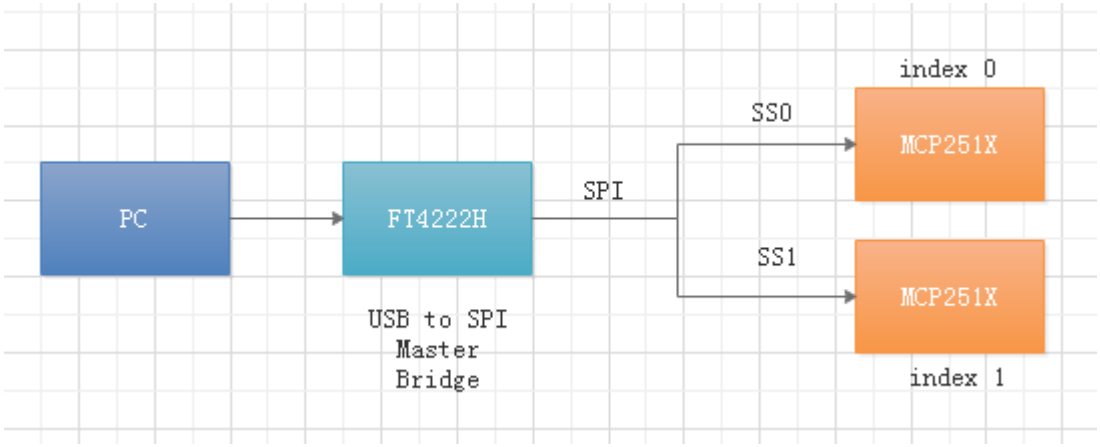

Figure 3.1 Sample Block Diagram

3.2 Code Listing

```
//------------------------------------------------------------------------------
//include Standard libraries
#include <windows.h>

#include <stdio.h>
#include <stdlib.h>
#include <vector>
#include <string>

//------------------------------------------------------------------------------
// include FTDI libraries
//
#include "ftd2xx.h"
#include "LibFT4222.h"


std::vector< FT_DEVICE_LIST_INFO_NODE > g_FT4222DevList;

//------------------------------------------------------------------------------
inline std::string DeviceFlagToString(DWORD flags)
{
	std::string msg;
	msg += (flags & 0x1) ? "DEVICE_OPEN" : "DEVICE_CLOSED";
	msg += ", ";
	msg += (flags & 0x2) ? "High-speed USB" : "Full-speed USB";
	return msg;
}

void ListFtUsbDevices()
```

```
{
	FT_STATUS ftStatus = 0;

	DWORD numOfDevices = 0;
	ftStatus = FT_CreateDeviceInfoList(&numOfDevices);

	for (DWORD iDev = 0; iDev < numOfDevices; ++iDev)
	{
		FT_DEVICE_LIST_INFO_NODE devInfo;
		memset(&devInfo, 0, sizeof(devInfo));

		ftStatus = FT_GetDeviceInfoDetail(iDev, &devInfo.Flags, &devInfo.Type,
&devInfo.ID, &devInfo.LocId,
			devInfo.SerialNumber,
			devInfo.Description,
			&devInfo.ftHandle);

		if (FT_OK == ftStatus)
		{
			printf("Dev %d:\n", iDev);
			printf("  Flags= 0x%x, (%s)\n", devInfo.Flags,
DeviceFlagToString(devInfo.Flags).c_str());
			printf("  Type= 0x%x\n", devInfo.Type);
			printf("  ID= 0x%x\n", devInfo.ID);
			printf("  LocId= 0x%x\n", devInfo.LocId);
			printf("  SerialNumber= %s\n", devInfo.SerialNumber);
			printf("  Description= %s\n", devInfo.Description);
			printf("  ftHandle= 0x%x\n", devInfo.ftHandle);

			const std::string desc = devInfo.Description;
			if (desc == "FT4222 A" || desc == "FT4222 B" || desc == "FT4222 D")
			{
				g_FT4222DevList.push_back(devInfo);
			}
		}
	}
}


int FT4222H_SPI_Init(int index, FT_HANDLE * ftHandle) {
	FT_STATUS ftStatus;
	int ret = 0;

	ftStatus = FT_Open(index, ftHandle);
	if (ftStatus != FT_OK) {
		printf("FT_Open failed (error code %d)\n", (int)ftStatus);
		ret = -1;
	}
	//for example, if you use port 0 and port 1, you need to set enable it with 0x02 +
0x01, because it is a global variable in IC
	FT4222_SPIMaster_Init(*ftHandle, SPI_IO_SINGLE, CLK_DIV_16, CLK_IDLE_LOW,
CLK_LEADING, 0x03);

	return ret;
}

int FT4222H_UnInit(FT_HANDLE ftHandle) {
```

```
	FT4222_UnInitialize(ftHandle);
	//ms = getms();
	FT_Close(ftHandle);

	return 0;
}
// To Read MX25L6435E ID
// The first is command 0x9F
// Then read the Manufacturer ID of 1-byte and followed by Device ID of 2-byte
void Read_MXIC_RDID(FT_HANDLE ftHandle)
{
	uint8 cmd = 0x9F;
	uint8 readData[3];
	uint16 sizeTransferred;

	// write one byte, command is 0x9F
	FT4222_SPIMaster_SingleWrite(ftHandle, &cmd, 1, &sizeTransferred, false);

	// then read 3 bytes
	FT4222_SPIMaster_SingleRead(ftHandle, &readData[0], 3, &sizeTransferred, true);

	printf("Manufacturer ID = %x\r\n", readData[0]);
	printf("Device ID = %x %x\r\n", readData[1], readData[2]);

}

//------------------------------------------------------------------------------
// main
//------------------------------------------------------------------------------
int main(int argc, char const *argv[])
{

	ListFtUsbDevices();

	if (g_FT4222DevList.empty()) {
		printf("No FT4222 device is found!\n");
		return 0;
	}
	// handle port 0 , spi communicate with first spi slave
	FT_HANDLE ftHandle1 = NULL;
	// handle port 1 , spi communicate with second spi slave
	FT_HANDLE ftHandle2 = NULL;

	FT4222H_SPI_Init(0, &ftHandle1);
	FT4222H_SPI_Init(1, &ftHandle2);

	// you can control spi communication with ftHandle1, ftHandle2
	Read_MXIC_RDID(ftHandle1);
	Read_MXIC_RDID(ftHandle2);

	FT4222H_UnInit(ftHandle1);
	FT4222H_UnInit(ftHandle2);

	return 0;

}
```

4 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited
(USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited
(Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited
(China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

http://ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Document References

FT4222H Product Page

LibFT4222

Acronyms and Abbreviations

Terms	Description
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
API	Application Programming Interface

Appendix B – List of Tables and Figures

List of Tables

List of Figures

Appendix C – Revision History

Document Title: AN_447 FT4222H Communication with Multi SPI Slaves

Document Reference No.: FT_001460

Clearance No.: FTDI#581

Product Page: https://ftdichip.com/products/ft4222h/

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	27-07-2023