

## **FT4222H support lib 1.4.8 - Revision Comments**

The files included in a standard support lib release are:

- D2XX Interface
  - ftd2xx.h
  - Dynamic library (dll directory)
    - i386\ftd2xx.lib (32-bit)
    - amd64\ftd2xx.lib (64-bit)
    - arm64\ftd2xx.lib (64-bit)
  - Static library (lib directory)
    - i386\ftd2xx.lib (32-bit)
    - amd64\ftd2xx.lib (64-bit)
- FT4222H Interface
  - LibFT4222.h
  - Dynamic library (dll directory)
    - i386\LibFT4222.dll (32-bit)
    - i386\LibFT4222.lib (32-bit)
    - amd64\LibFT4222-64.dll (64-bit)
    - amd64\LibFT4222-64.lib (64-bit)
    - arm64\LibFT4222-64.dll (64-bit)
    - arm64\LibFT4222-64.lib (64-bit)
  - Static library (lib directory)
    - vs2013\i386\LibFT4222.lib (32-bit)
    - vs2013\amd64\LibFT4222-64.lib (64-bit)
    - ucrt\i386\LibFT4222.lib (32-bit)
    - ucrt \amd64\LibFT4222-64.lib (64-bit)
- Sample codes
  - In Sample folder, LibFT4222\_examples.sln can be used to browser all samples.
  - This Sample code is linked to Dynamic library

On making a new release the files will be posted on FTDI's web site  
<http://www.ftdichip.com/Products/ICs/FT4222H.html> in ZIP file format.

The Libft4222 dynamic libraries in i386 and amd64 are based on Microsoft Visual C++ 2015 x86 and x64 Redistributable package. The ARM64 version is based on the Visual C++ 2022 Redistributable that supports Windows devices with ARM64 architecture. If you want to build up a dynamic link executable file, please install the below package on the target machine.

<https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170>

For building up a static link executable file, libft4222 supports Visual Studio 2013 and Visual Studio which version is 2015 or advance. If the version of Visual Studio is equal to or greater than 2015, you need to link with the library in UCRT folder.

All the examples are built with dynamic link except "getting\_started\_static" project. The "getting started\_static" is built with static linking and does not support ARM64.

Starting from v1.4.6, the calling convention of our library has been changed to "stdcall". Here are the notes on usage.

- If there is a calling convention mismatch between the application and the DLL, recompiling the application is necessary to ensure that it understands and passes function parameters correctly.
- If your x86 application is written in other languages (e.g. C#, Delphi, etc.) and calls C++ DLLs via P/Invoke or other mechanisms, then you need to ensure that the correct calling convention is used in those languages as well.

**Release version**

1.4.8 (JUNE 30, 2025)

Maintenance release.

**Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Add ARM64 dynamic library and examples.

**Release version**

1.4.7 (October 30, 2024)

Maintenance release.

**Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Fix csharp\_spi\_master complier issue.
  - Fix FT4222\_GetVersion issue.
  - Add x64 examples.

**Release version**

1.4.6.1 (April 3, 2024)

Maintenance release.

**Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Fix dll file does not be signed issue.

**Release version**

1.4.6 (March 7, 2024)

Maintenance release.

**Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Add static lib support for VS2013 and the version is higher than VS2015(UCRT).
  - Add API FT4222\_GetChipMode.
  - Calling convention is changed to "stdcall".

**Release version**

1.4.5 (June 14, 2022)

Maintenance release.

**Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Remove libboost linking.
  - Add Feature. Add API FT4222\_SPIMaster\_SetMode. It can change the polarity and phase
  - Change the member names in enum SPI\_ChipSelect
  - Add event trigger in interrupt example

- Microsoft Visual C++ 2010 runtime library is already built in

### **Release version**

1.4.4 (Nov 12, 2020)

Maintenance release.

### **Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Fix issue. Frequency 100K~400K in I2C Master is not accurate.
  - Add Feature. Add API FT4222\_I2CMaster\_ResetBus. This function can reset i2c bus when it is abnormal.
  - Add Feature. Add API FT4222\_SPIMaster\_SetCS. It can determine the CS is high or low when SPI bus is active. The default value is active-low.
  - Remove Feature. Remove API FT4222\_SPISlave\_RxQuickResponse. It may cause data lost sometimes.
  - Add Feature. Add license announce in libft4222.h.
  - Add Feature. Add static library, need to define FT4222\_STATIC during building.

### **Release version**

1.4.3 (Mar 19, 2020)

Maintenance release.

### **Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Fix issue. FT4222\_I2CMaster\_WriteEx does not return correct sizeTransferred.
  - Fix issue. FT4222\_SPIMaster\_MultiReadWrite does not return correct sizeofRead when multiReadBytes equal to zero.

### **Release version**

1.4.2 (Dec 5, 2018)

Maintenance release.

### **Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Function changed. Now I2c master can send larger than 512 bytes.
  - Function changed. Adopt FT\_SetTimeouts to be the default timeout of FT4222\_I2CMaster\_Read.

### **Release version**

1.4.1 (Mar 15, 2018)

Maintenance release.

### **Release Fixes**

- Support Chip version "A", "B", "C" and "D".
  - Fix FT4222\_GetClock get error issue.
  - Fix issue. Repeated\_START does not work in API.

- FT4222\_I2CMaster\_WriteEx.
- Fix issue. FT4222\_UnInitialize hang on linux platform in SPI Slave Mode.
- Fix issue. SPI slave cannot receive more than 8192 bytes in SPI with protocol mode.
- FT4222\_I2CMaster\_Init can support speed < 60K bps.
- Add API FT4222\_ChipReset.
- Add API FT4222\_SPISlave\_SetMode.
- Add API FT4222\_GPIO\_SetWaveFormMode(only available in Chip Rev D).
- Add API FT4222\_SPISlave\_RxQuickResponse (only available in Chip Rev D).
- Add SPI flash access examples.
- Add SPI Slave sample(NO protocol, Master/Slave side).

### **Release version**

1.4 (May 26, 2017)

Maintenance release.

### **Release Fixes**

- Support Chip version "A", "B" and "C".
- Fix GPIO as interrupt sampling issue. Now trigger times can be observed in every fetch.
- Fix performance issue on SPI Slave Receiving and GPIO receiving.
- FT4222\_I2CMaster\_WriteEx can send 65531 bytes in one transition (previous is 508 bytes)

### **Release version**

1.3 (Aug 8, 2016)

Maintenance release.

### **Release Fixes**

- Support Chip version "A", "B" and "C"
- Add API FT4222\_I2CSlave\_SetClockStretch/FT4222\_I2CSlave\_SetRespWord.
- Add API FT4222\_SPISlave\_Init\_Ex
- SPI Master provide CLK\_2 option.
- interrupt status can be read by FT4222\_GPIO\_Read and FT4222\_GPIO\_ReadTriggerQueue

### **Release version**

1.2 (Aug 24, 2015)

Maintenance release.

### **Release Fixes**

- Support Chip version "A" and "B"
- add API I2CMaster ReadEx/WriteEx/GetStatus

### **Release version**

1.1 (Sep 4, 2014)

Maintenance release.

**Release Fixes**

- Support Chip version "A"