NuTool-USB to Serial Port
User Manual

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

www.nuvoton.com

Table of Contents

List of Figures

1 Overview

NuTool-USB to Serial Port is a host-side software, it passes through and montiors I²C, SPI and CAN data of Nu-Link2-Pro® adapter. Nu-Link2-Pro® adapter supports various bridge applications including USB-RS485, USB-I²C, USB-SPI and USB-CAN Bridge. (Only USB-I²C, USB-SPI and USB-CAN are supported in NuTool-USB to Serial Port. For USB-RS485, user can use any off-the-shelf serial port terminal, e.g. Tera term) In the following chapter, the word Nu-Link2-Bridge will be used to represent the pass-through bridge application on Nu-Link2-Pro® adapter. All the bridges application acted as a general USB communications device class (USB CDC), so user can access the Nu-Link2-Bridge with many programming language or with any third party terminal software. The following chapter will focus on using the NuTool-USB to Serial Port to access Nu-Link2-Bridge.

About the wire connections of Nu-Link2-Bridge, please refer to chapter 3.3 of UM_Nu-Link2-Pro_Debugger_and_Programmer_User_Manual.

The source code of Nu-Link2-Bridge firmware and NuTool-USB to Serial Port can be found in the following repositories hosted by Nuvoton.

https://github.com/OpenNuvoton/Nu-Link2-Bridge_Firmware

https://github.com/OpenNuvoton/NuTool-USB-to-Serial-Port

1.1 About Nu-Link2-Bridge Firmware

The Nu-Link2-Bridge firmwares are based on the HSUSBD_VCOM_SerialEmulator which is a ready to use USB-UART Bridge. The UART part can be replaced with other desired interface to become a new bridge application. The desired interface includes RS485, I²C, SPI and CAN. Since the USB part is unchanged and follows the specification defined by USB-IF, so the Nu-Link2-Bridge can benefit from any existing resource. For example, user can use pip command to install pyserial for python language, or use npm command to install serialport for node.js to access Nu-Link2-Bridge.

1.2 About NuTool-USB to Serial Port

The NuTool-USB to Serial Port is based on the Terminal Example of Qt. Qt is a cross-platform application development framework for desktop, embedded and mobile devices. Supported Platforms include Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS and others. By using QT, the NuTool-USB to Serial Port is a free software under GPL license.

2 Using NuTool-USB to Serial Port in Windows

Since Qt is a cross-platform framework, user can compile this sample for Linux, OSX and Windows without any further modification. We supply Windows executable file by default.

2.1 Main Screen

When the NuTool-USB to Serial Port starts, a main screen will be displayed as shown in Figure 2-1. The main screen is divided into three parts. First, user can issue connect, disconnect or clear log messages from the Menu or tool bar. Besides the menu and tool bar, there are two frames for sending and receiving data. If Nu-Link2-Bridge is configured as input only mode like CAN Silent mode, I²C Monitor, or SPI Monitor mode, only the Received message frame is shown.

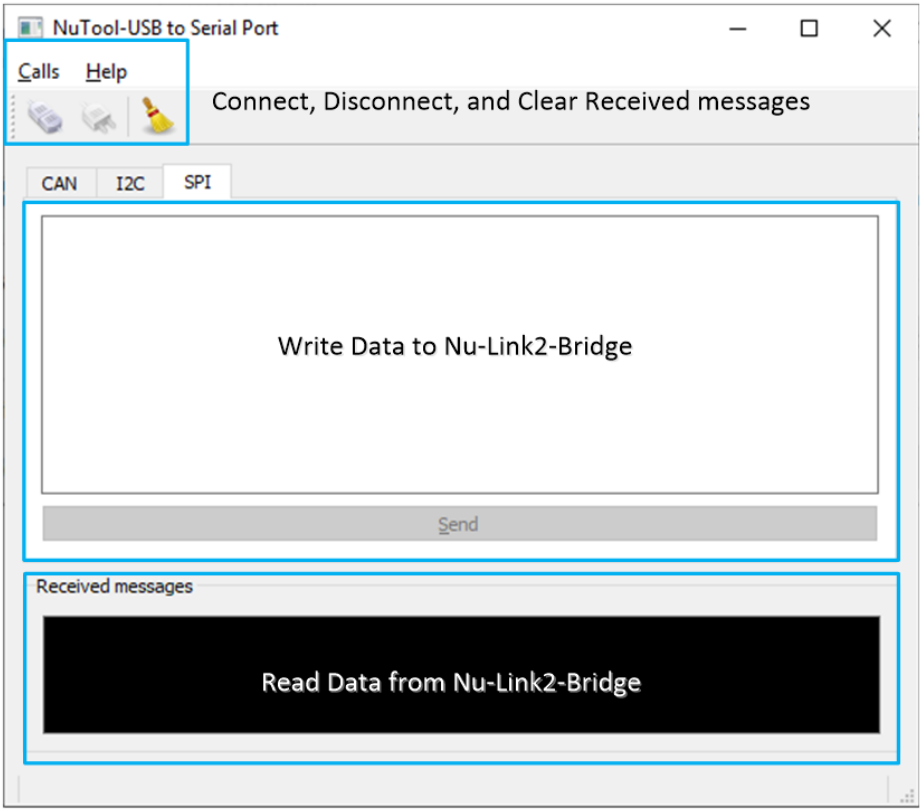

Figure 2-1 Main Screen

2.2 Connect Settings

Before connecting to the Nu-Link2-Bridge, please make sure the dedicated bridge application is already programmed or already built in the Nu-Link2-Pro adapter. When user issues connect from the menu or from the tool bar, a setting window pops up as shown in Figure 2-2. In the left part, user should select the serial port for Nu-Link2-Bridge. In the right part, user can configure the setting for each interface. After the setting is ready, click **OK** to connect to the Nu-Link2-Bridge or click **Cancel** to abort connection. In the bottom, user can log the received data from Nu-Link2-Bridge to a text file.

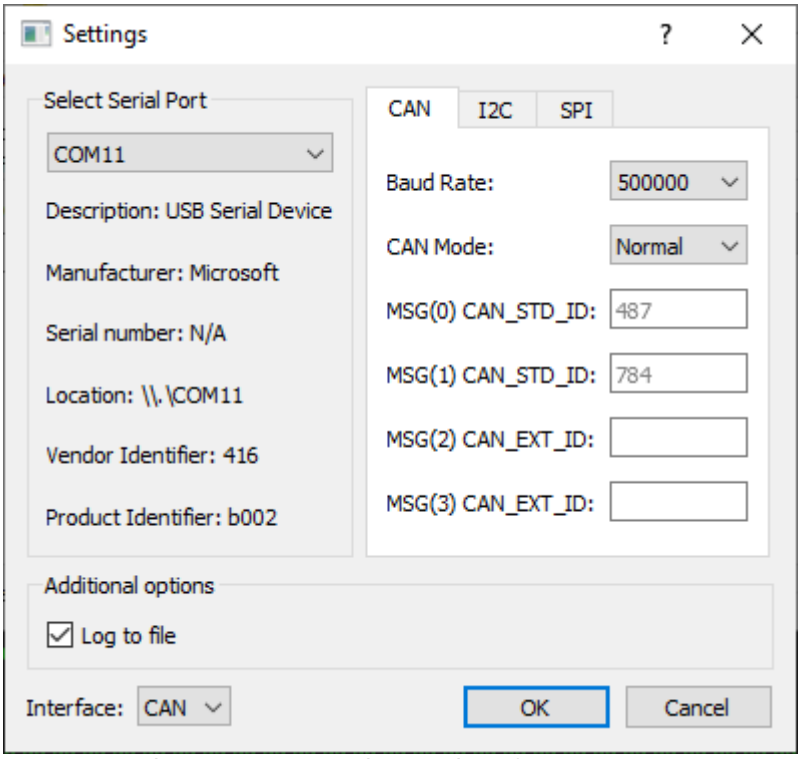

Figure 2-2 Connection Settings for CAN

2.2.1 CAN Settings

The Baud Rate setting must be consistent with the connecting network. Also, USB-CAN Bridge can be configured as Normal mode or Silent mode due to user's purpose.

In Normal mode, user can listen to four CAN devices at most in the same CAN bus. The first two objects are reserved for CAN_STD_ID and others are reserved for CAN_EXT_ID. In such mode, user can also send a frame to any CAN device, which can be set in the main screen.

In Silent mode, USB-CAN Bridge can monitor any data from any CAN device in the same network. This mode is used to debug the CAN Bus communication.

2.2.2 I²C Settings

The USB-I²C Bridge can be set to I²C Master with specific bus clock. In this mode, it behaves as an I²C Master. It can also be set to Monitor mode, I²C Monitor mode is a special feature of the USCI_I²C design. It is actually an I²C slave variation. The USB-I²C Bridge can read any data in the I²C bus without affecting the bus in Monitor mode. The sending frame in main screen is hidden in I²C Monitor mode.

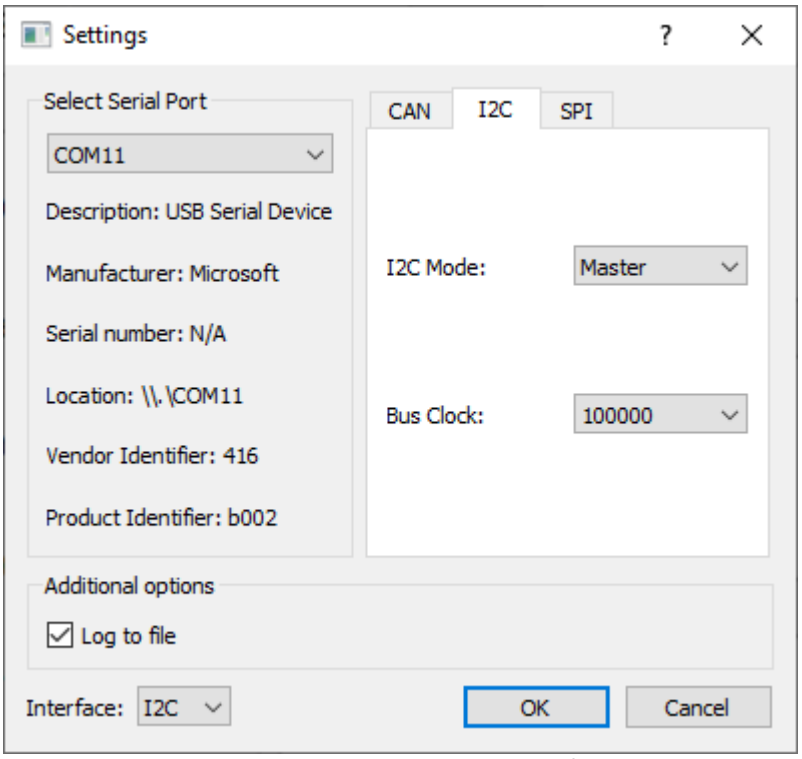

Figure 2-3 Connection Settings for I²C

2.2.3 SPI Settings

All the SPI setting options are relative to the SPI waveform. Please refer to the Technical Reference Manual for further details. The SPI Monitor mode is implemented by combination of two SPI Slave without MISO pin, thus it won't output any data. It is recommended to tie as many ground pins between the target device and USB-SPI Bridge to reduce noise if you found the USB-SPI Bridge will affect the SPI bus.

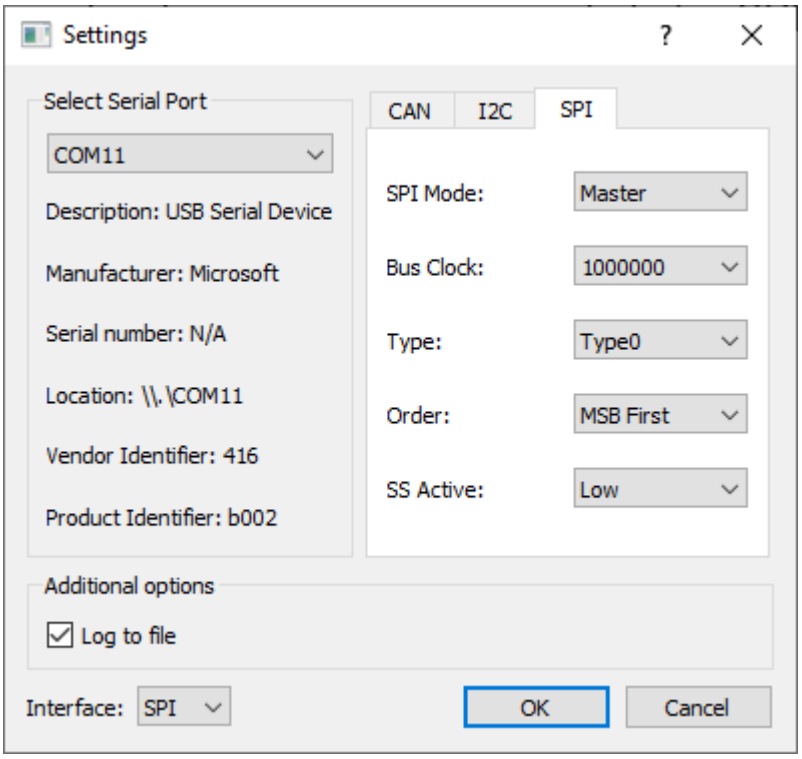

Figure 2-4 Connection Settings for SPI

2.2.4 Log File

If the “Log to file” option is selected, there will be a text file named “LogData.txt” generated automatically. All the raw data from Nu-Link2-Bridge is stored in this file. The NuTool-USB to Serial Port converts the raw data from binary to hexadecimal characters, so it can be opened by any text editor like notepad++. Please note that, the timestamp in the log file is generated by NuTool-USB to Serial Port, not Nu-Link2-Bridge and the data length is the read length by NuTool-USB to Serial Port. The read length may not equal the real output length by Nu-Link2-Bridge.

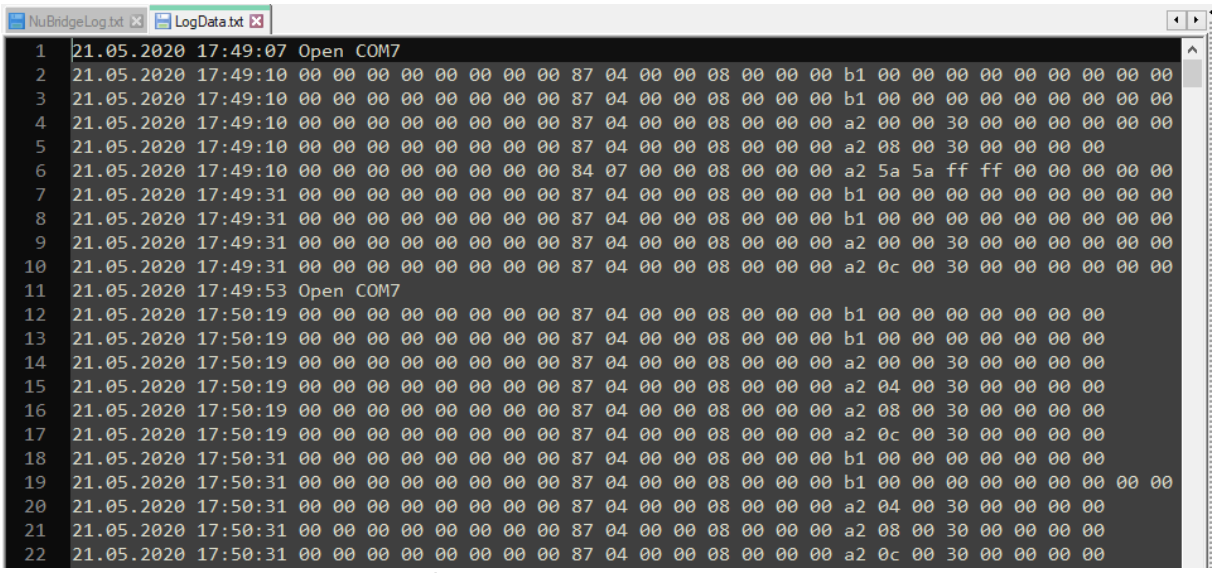

Figure 2-5 LogData.txt

2.3 USB-CAN Bridge

User can send and receive data in CAN normal mode, and can only receive data in CAN silent mode.

In CAN normal mode, user must set frame type correctly and specifies the Frame ID and 8-Byte Payload before clicking Send button.

In CAN silent mode, it displays all data from all CAN devices in the same network. The structure of logged data is: ID Type (4 bytes) Frame Type (4 bytes) Message ID (4 bytes) Data length (1 byte) Data (8 bytes) padding (3 bytes).

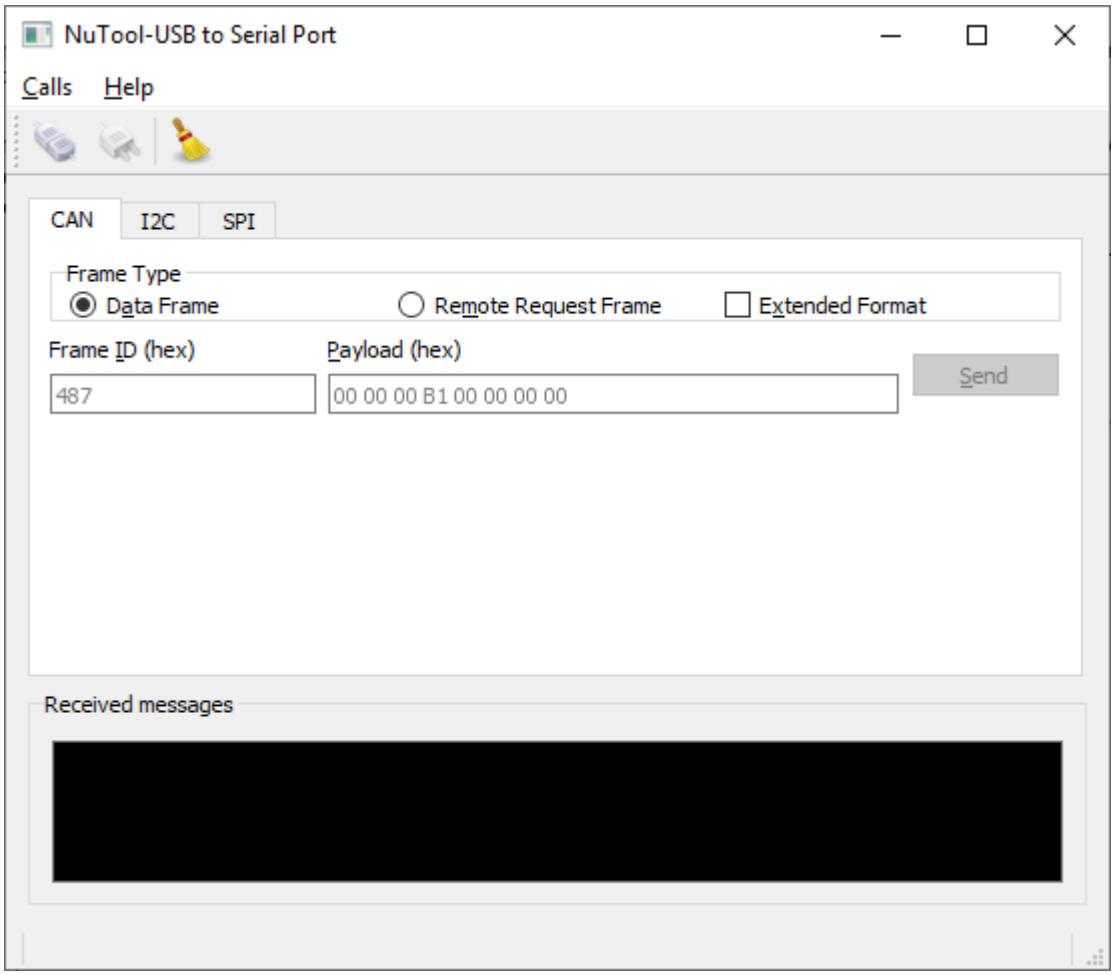

Figure 2-6 USB-CAN Bridge Main Screen

2.4 USB-I²C Bridge

The normal mode supports I²C Read and I²C Write operations and only 7-bit I²C address is supported. After I²C Read Address and Read Length is sent, the received read data will be displayed in the Received messages frame. After I²C write triggered, terminal will show 0x4F, 0x4B (means “OK” in ASCII) or 0x4E, 0x47 (means “NG” in ASCII).

In I²C Monitor mode, it reads all data from all I²C devices in the same I²C bus. It monitors the raw data plus I²C control bit, the format is shown as follows: I²C Start bit (0x53, 0x00 in log), I²C Stop bit (0x50, 0x00 in log), I²C ACK and DATA (0x41, data_byte in log) and I²C NACK and DATA (0x4E, data_byte in log).

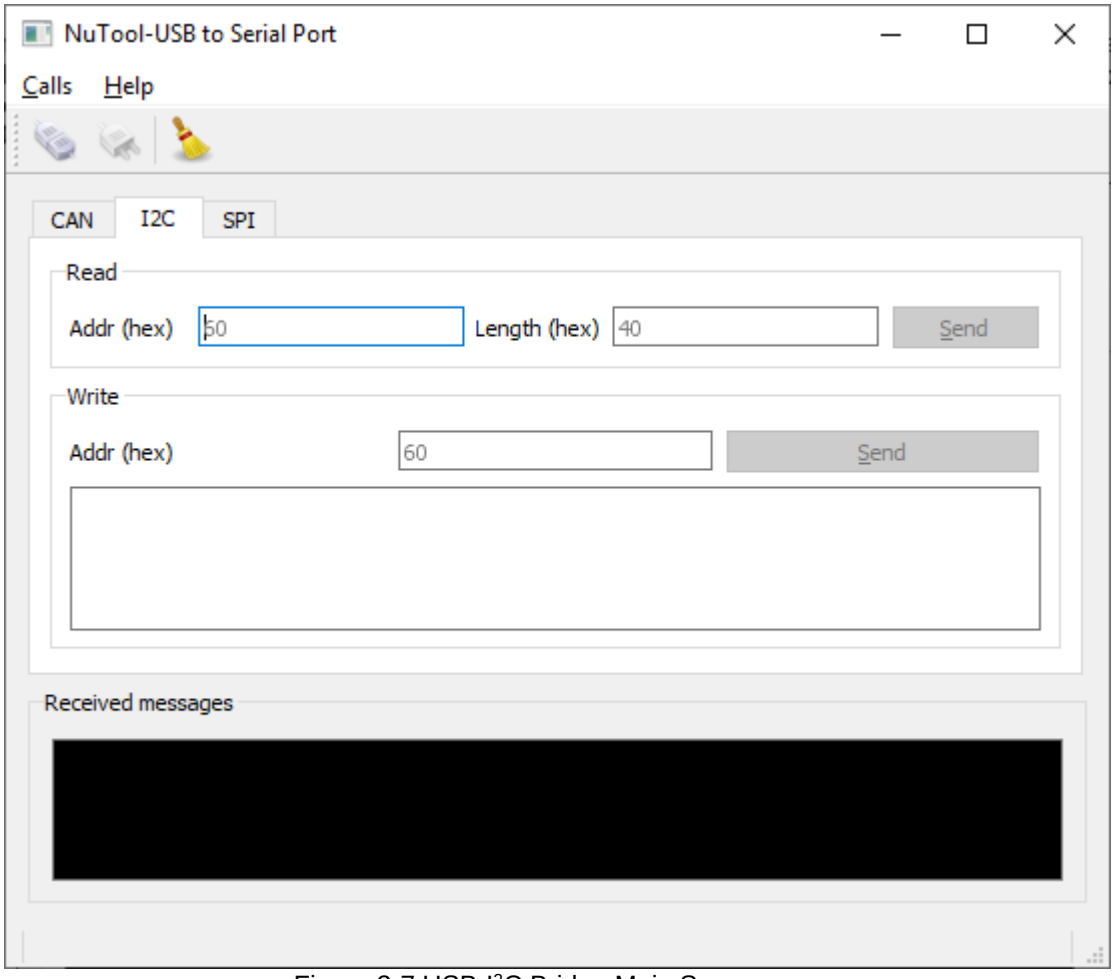

Figure 2-7 USB-I²C Bridge Main Screen

2.5 USB-SPI Bridge

Since SPI is a Full-duplex synchronous interface, USB-SPI Bridge only supports one I/O operation. After filling in the data and clicking the Send button, all the data are sending out through SPI MOSI pin and some data coming in at the same time through SPI MISO pin. The received data is displayed in the Received messages frame.

In SPI Monitor mode, two SPI Slaves are used to monitor the SPI bus data. Since the data is logged from two different pins, it displays interleaving but not continuous. That is to say, the odd byte data comes from MOSI and the even byte data comes from MISO.

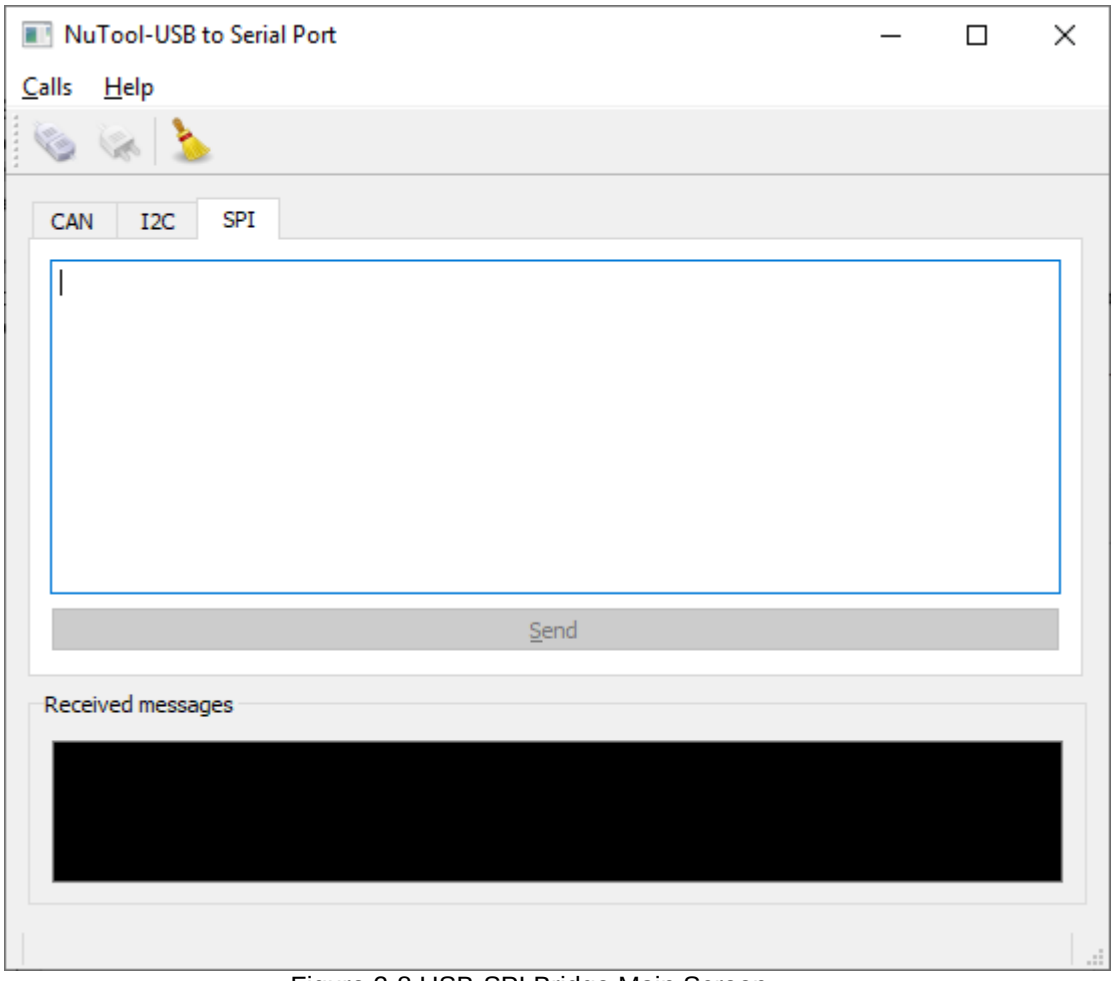

Figure 2-8 USB-SPI Bridge Main Screen

3 Revision History

Date	Revision	Description
2020.12.10	1.00	First version.

Important Notice

Using this software indicates your acceptance of the disclaimer hereunder:

THIS SOFTWARE IS FOR YOUR REFERENCE ONLY AND PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. YOUR USING THIS SOFTWARE/FIRMWARE IS BASED ON YOUR OWN DISCRETION, IN NO EVENT SHALL THE COPYRIGHT OWNER OR PROVIDER BE LIABLE TO ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.