Visible Light Communications Demonstration: Creating and Demonstrating a Serial Connection using VLC Transceivers

Jimmy C. Chau <jchau@bu.edu>
Thomas Little <tdcl@bu.edu>

Multimedia Communications Laboratory
Electrical and Computer Engineering, Boston University
8 Saint Mary's Street, Room 324
Boston, MA 02215-2421
Phone: +1 (617) 353-9877

Smart Lighting Engineering Research Center*

MCL Technical Report #02-15-2010

*This work is supported by the NSF under grant No. EEC-0812056. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

Abstract

In order to demonstrate the feasibility of integrating optical communications into a lighting system, this demonstration kit includes a pair of optical transceivers that produce bright white light. This demonstration shows visible light communication at nearly 1Mbps[1]. Instructions for setting up this demonstration are provided by this manual.

1 Overview

The visible light communication (VLC) transceiver provided in this demonstration was designed to transmit Manchester-encoded data at 1Mbps over an optical channel while simultaneously providing illumination. However, the transceiver can also support the simpler non-return-to-zero (NRZ) encoding given that transitions between high and low states occur frequently.

The VLC demonstration documented herein shows data transmission from one serial terminal through a VLC channel to another serial terminal. Each serial terminal consists of a computer, the demonstration software, and a USB-to-serial converter.

Since computer serial ports natively use NRZ instead of Manchester encoding, the demonstration software repeatedly sends the chosen message over the serial port to ensure that frequent transitions between low and high states occur.

2 Software installation

To support multiple operating systems, the demonstration software is written as a Java application; RXTX is used to provide Java with support for serial communications. In order to use the demonstration software, you will need to install a Java Runtime Environment (JRE), the driver for the USB-to-serial converter, and RXTX.

2.1 Java Runtime Environment installation

The Java Runtime Environment, which is often simply called Java, is needed to run Java applications such as the one used in this demonstration. On many computers, a JRE is already installed and another installation is not necessary.

However, if a JRE is not already installed, instructions for obtaining and installing one can be found at http://java.com/.

2.2 RXTX installation

RXTX is required to allow the demonstration software to access the computer's serial port. To install RXTX, follow the instructions below for the appropriate operating system. The files required for installing RXTX are provided on this disk in "`SupportingSoftware/rxtx-2.1-7-bins-r2/`".

2.2.1 For Windows

1. Locate the folder where your JRE was installed. Likely locations include the following:
   - `C:\Program Files (x86)\Java\jre6\`
   - `C:\Program Files\Java\jre6\`
   - `C:\Program Files\Java\jre1.6.0_01\`

[1]This demonstration sets the serial ports to 921600 baud.

Note that depending on when you installed Java, the version numbers may be different.

From here on, that folder will be called <JavaHome>; when following these instructions, you should replace <JavaHome> with the actual location of the JRE installation.

2. Copy "SupportingSoftware\rxtx-2.1-7-bins-r2\Windows\i386-mingw32\rxtxParallel.dll" to "<JavaHome>\bin\".

3. Copy "SupportingSoftware\rxtx-2.1-7-bins-r2\Windows\i386-mingw32\rxtxSerial.dll" to "<JavaHome>\bin\".

4. Copy "SupportingSoftware\rxtx-2.1-7-bins-r2\RXTXcomm.jar" to "<JavaHome>\lib\ext\".

2.2.2 For Mac OS X

Simply copy "SupportingSoftware\rxtx-2.1-7-bins-r2\Max_OS_X\librxtxSerial.jnilib" and "SupportingSoftware\rxtx-2.1-7-bins-r2\RXTXcomm.jar" to "/Library/Java/Extensions" or "~/Library/Java/Extensions".

For more information, see http://rxtx.qbang.org/wiki/index.php/Installation_on_MacOS_X.

2.2.3 For Linux

Some Linux distributions have RXTX available through a software repository; the name may simply be "rxtx" or it may be more complicated (e.g., "librxtx-java"). If RXTX is available this way, simply use the software management system to install RXTX.

If RXTX is not available through your operating system's software repositories, there exist multiple ways to install RXTX; see http://rxtx.qbang.org/wiki/index.php/Installation_on_Linux and "INSTALL" in "SupportingSoftware/rxtx-2.1-7-bins-r2/" on this disk for more information.

2.3 USB-to-serial converter driver installation

The USB-to-serial converter provided with this demonstration is a FTDI TTL-232R USB to TTL Serial Converter Cable. To use this USB-to-serial converter for this demonstration, its Virtual COM port (VCP) driver must be installed. This VCP driver may already be installed on your computer.

The latest drivers for and more information about the FTDI USB-to-serial converter cables can be found at the FTDI website: http://www/ftdichip.com. Installation instructions for Windows, Mac, and Linux operating systems are provided below.

2.3.1 VCP driver installation for Windows

For Windows 2000, XP, Vista, and 7: In addition to the VCP drivers, Windows 2000 through Windows 7 also requires FTDI's D2XX drivers. To install these drivers, simply run "CDM20600.exe"[2] in "Drivers\FTDI_VCP\Windows2000-7\" on this disk.

For Windows 98 and ME: Follow the instructions in "Windows_98_Installation_Guide.pdf" in "Drivers\FTDI_VCP\Windows98-ME\" on this disk. Note that the drivers are included on this disk in "Drivers\FTDI_VCP\Windows98-ME\R10906".

2.3.2 VCP driver installation for Mac

For Mac OS X: Open "FTDIUSBSerialDriver_v_2_2_14.dmg" in "Drivers/FTDI_VCP/MacOSX". Then install the package that matches the computer's version of Mac OS X; for example, if the computer uses Mac OS X 10.4, install "FTDIUSBSerialDriver_10_4_10_5_10_6.mpkg".

[2]If your PDF reader supports this feature, clicking this link will launch "CDM20600.exe" for you.

For other versions of Mac OS: See the FTDI drivers webpage at `http://www.ftdichip.com/Drivers/VCP.htm` for more information.

2.3.3 VCP driver installation for Linux

The "ftdi_sio" driver is required to use the FTDI USB-to-serial converter in Linux. On many Linux operating systems, such as Ubuntu 9.10, the ftdi_sio driver is already installed. In this case, reinstalling the driver should not be necessary.

To check if ftdi_sio driver is installed:

1. Open a terminal to enter commands.
2. Plug in the FTDI USB-to-serial converter cable.
3. Enter either of the following commands:

   ```
   dmesg | grep -F "ftdi_sio"
   lsmod | grep -F "ftdi_sio"
   ```

4. Check to see if "ftdi_sio" is shown in either of the outputs. If it is shown, then the ftdi_sio driver is installed.

If ftdi_sio is not already installed: You can install it through one of the following methods:

- Follow the instructions within "README.txt" in "`Drivers/FTDI_VCP/Linux/`" on this disk;
- If the ftdi_sio driver is available through your operating system's software management system, use the software management system to install the driver;
- Or, enable the `USB_SERIAL_FTDI_SIO` option in the Linux kernel's configuration; then rebuild and reinstall the Linux kernel.

3 Hardware setup

The hardware is mostly setup since the transceiver are fully assembled. However, some steps remain to power the transceiver and to connect it to the computer.

3.1 Parts needed

The following parts are needed to setup the hardware for this demonstration.

- 2 transceivers,
- 2 FTDI USB-to-serial converter cables,
- 2 power supplies with 2 power cords,
- and 4 jumper wires.

These parts are shown in figure 1 on the following page and should be included in the demonstration kit. Two computers[3], each with a USB port, to run the demonstration software are also needed.

[3]The two computers are not included in the demonstration kit.

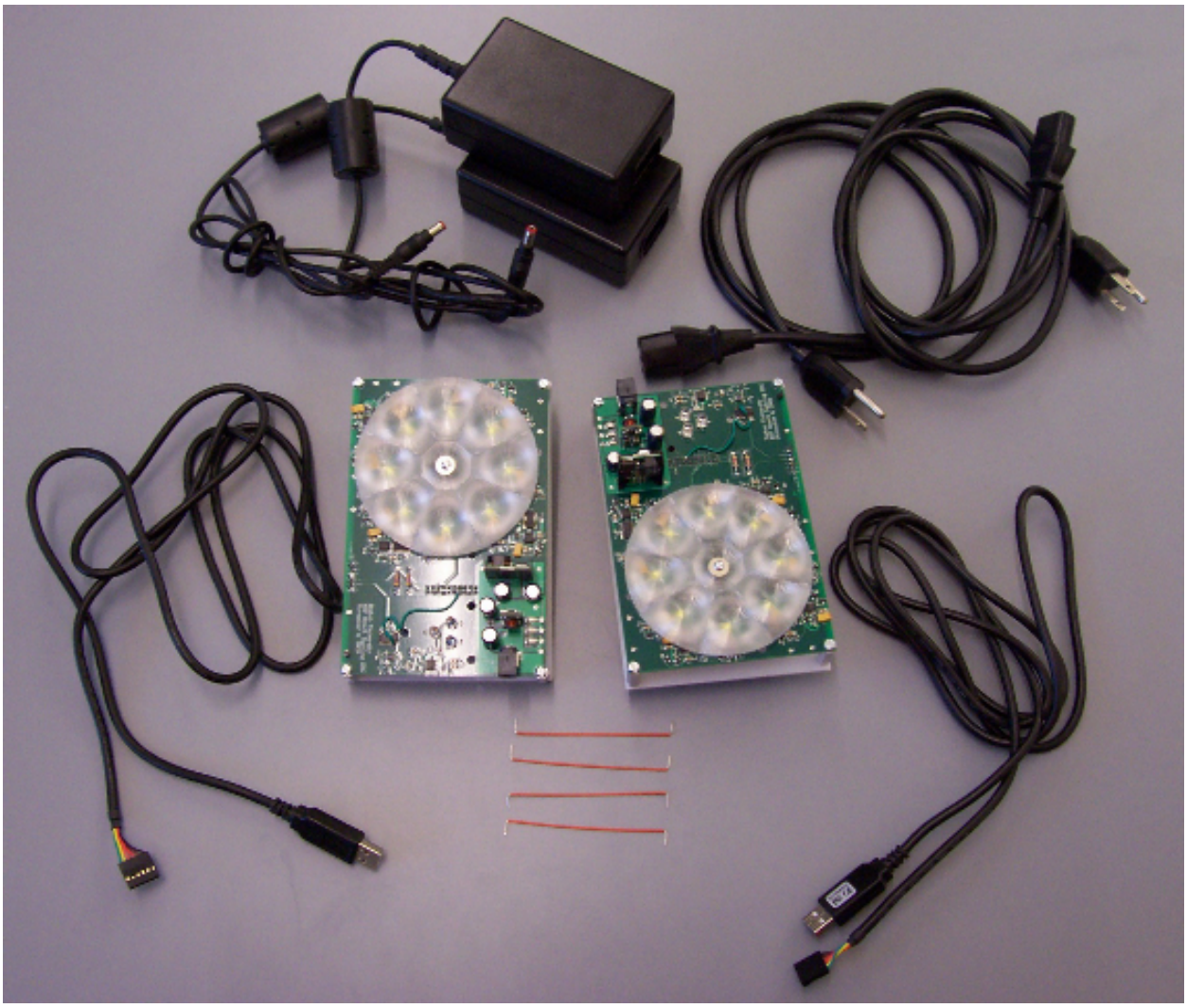

Figure 1: The hardware provided in the demonstration kit is shown. The two transceivers are shown in the center. Going clockwise, starting from the top left, are the two power supplies, power cords, a USB-to-serial converter cable, jumper wires, and another USB-to-serial converter cable.

3.2 Connecting power

To supply the transceivers with power,

- plug the 12V DC power supply into an electrical outlet.
- Then connect the 12V power supply's DC output (the barrel connector) to the transceiver's 12V power jack.

When power is supplied to the transceiver, its +12V, +5V, and -5V indicator light-emitting diodes (LEDs) should all light.

3.3 Connecting data

Caution: The transceiver should be powered on before the transmitter input (TX_IN) data connection is made.

Select one transceiver and one computer to serve as the transmitter. The other transceiver and other computer will be the receiver.

3.3.1 To connect the transmitter

1. Connect one USB-to-serial converter cable to the transmitter's computer.
2. Connect one end of a jumper wire to a GND hole on the transmitting transceiver's female header[4].
3. Plug the other end of that GND jumper wire into GND of the transmitter's USB-to-serial converter cable[5].
4. Use another jumper wire to connect the TX_IN on the transmitting transceiver's female header to the TXD on the transmitter's USB-to-serial converter cable.

When the transmitter is connected, the white LEDs should turn on.

3.3.2 To connect the receiver

1. Connect the remaining USB-to-serial converter cable to the receiver's computer.
2. Use a jumper wire to connect the GND on the receiving transceiver to the GND on the receiver's USB-to-serial converter cable.
3. Use a jumper wire to connect the corrected receiver output (RX_OUT2) on the receiving transceiver to the RXD on the receiver's USB-to-serial converter cable.

4 Operation

After installing the required software and setting up the hardware, run the demonstration application on both the transmitting computer and the receiving computer.

Due to differences between JRE installations on Windows, Mac, and Linux computers, different steps are necessary to start the demonstration software. However, after starting the demonstration, operation of the demonstration across all operating systems is identical.

[4]For help finding the transceiver's female header, see figure 2 on the next page.
[5]For help identifying wires on the USB-to-serial converter cable, see figure 3 on page 9.

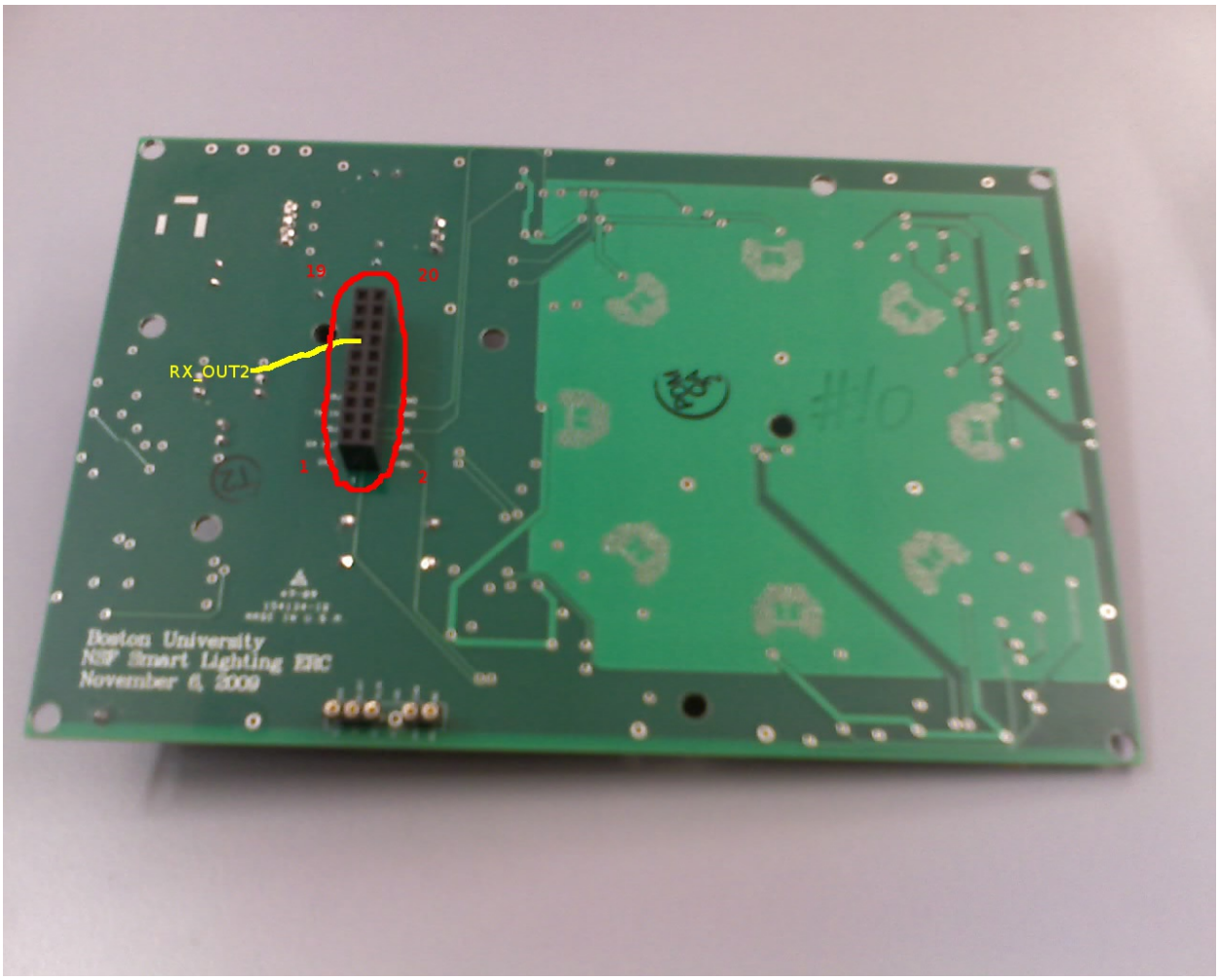

Figure 2: The rear of a transceiver board is shown with the "female header" circled in red. The lowest left pin hole in the female header is pin hole 1; the bottom-most right pin hole is pin 2; each of the other pin holes are numbered to be two greater than the one directly under it. Pin hole 13 is RX_OUT2 and is shown in yellow. All other connected pin holes are labeled in white text on the silkscreen layer.

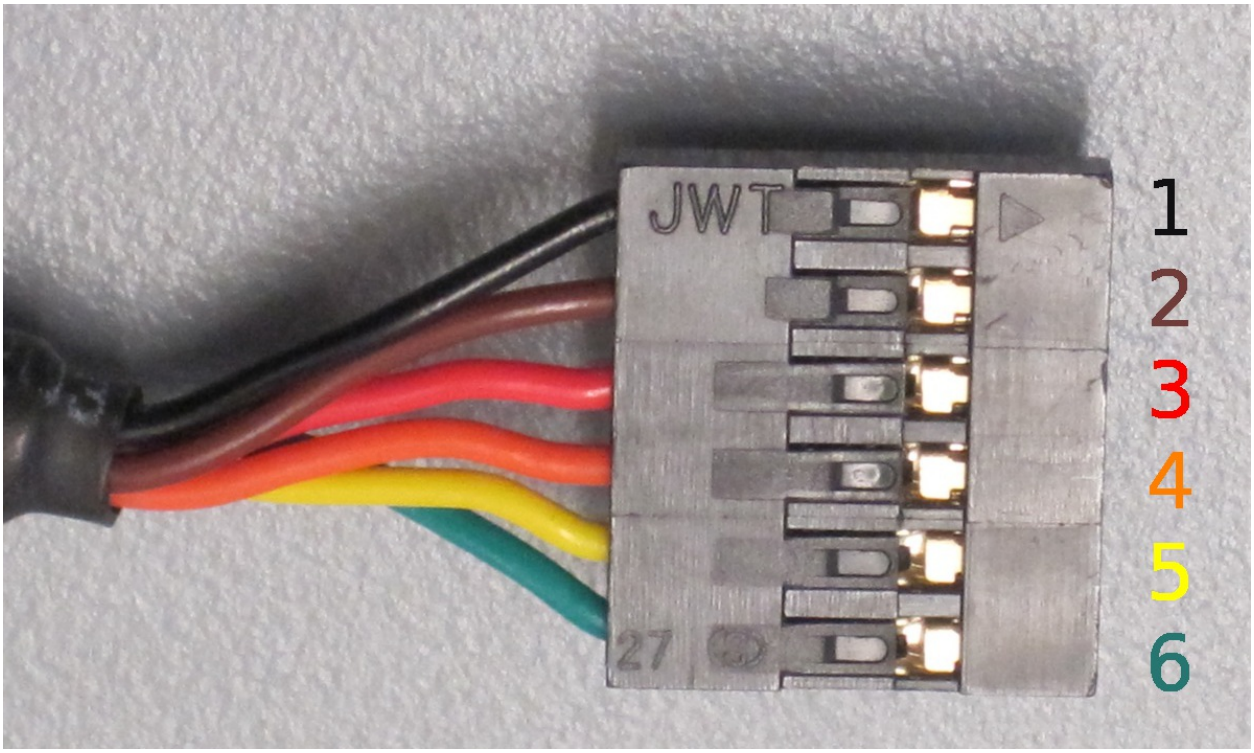

Figure 3: The serial end of the FTDI USB-to-serial converter cable is shown with the wires labeled. Wire 1 (black) is GND. Wire 4 (orange) is TXD. Wire 5 (yellow) is RXD.

4.1 Starting the demonstration application

4.1.1 On Windows

1. Recall the JRE installation location from section 2.2.1 on page 3. Once again, this manual will use `<JavaHome>` to refer to this location.
2. Start a command prompt or the "Run" application.
3. Enter the following command, where `<JavaHome>` refers to the installation location of the JRE and `<DemoDisk>` refers to the location of this disk (e.g., "`D:\`"):

   ```
   <JavaHome>\bin\java.exe -jar <DemoDisk>\VLCSerialPortDemo.jar
   ```

4.1.2 On Mac OS X or Linux

1. Open a terminal to enter commands.
2. Use "`cd`" to change directory to this demonstration disk.
3. Execute the following command to start the demonstration application:

   ```
   java -jar VLCSerialPortDemo.jar
   ```

4.2 Using the demonstration application

1. At this point, the graphical user interface (GUI) shown in figure 4 on the following page should appear.
2. Select the USB-to-serial converter cable's serial port. If more than one serial port appears on the list, you can use your operating system's utilities to determine which serial port corresponds to the USB-to-serial converter cable. Alternatively, to tell which serial port corresponds to the USB-to-serial converter cable you can unplug the USB-to-serial converter cable and restart the demonstration application to see which serial port disappears.
3. Click the "Open selected serial port" button.
4. At this point, the demonstration program should switch to the GUI shown in figure 5 on page 12.

4.2.1 On the receiving terminal

On the receiving terminal, the demonstration application will display the most recently received data until the application exits. The "Close" button may be used to exit the application.

4.2.2 On the transmitting terminal

Since the transmitting terminal does not receive data, it will display no data in the "Received" text area.

When the demonstration application starts, it automatically begins to repeatedly transmit a logo — featuring the Boston University mascot, Rhett— in text.

The data to be repeatedly transmitted can be changed by entering a new string into the "Send string:" field; the data can also be changed by choosing a file to send using the "Choose File..." button. A sample file containing the Rhett logo is provided as "rhett_ascii.txt" in "`data/`" on this disk.

To stop sending and quit the demonstration, click the "Close" button.

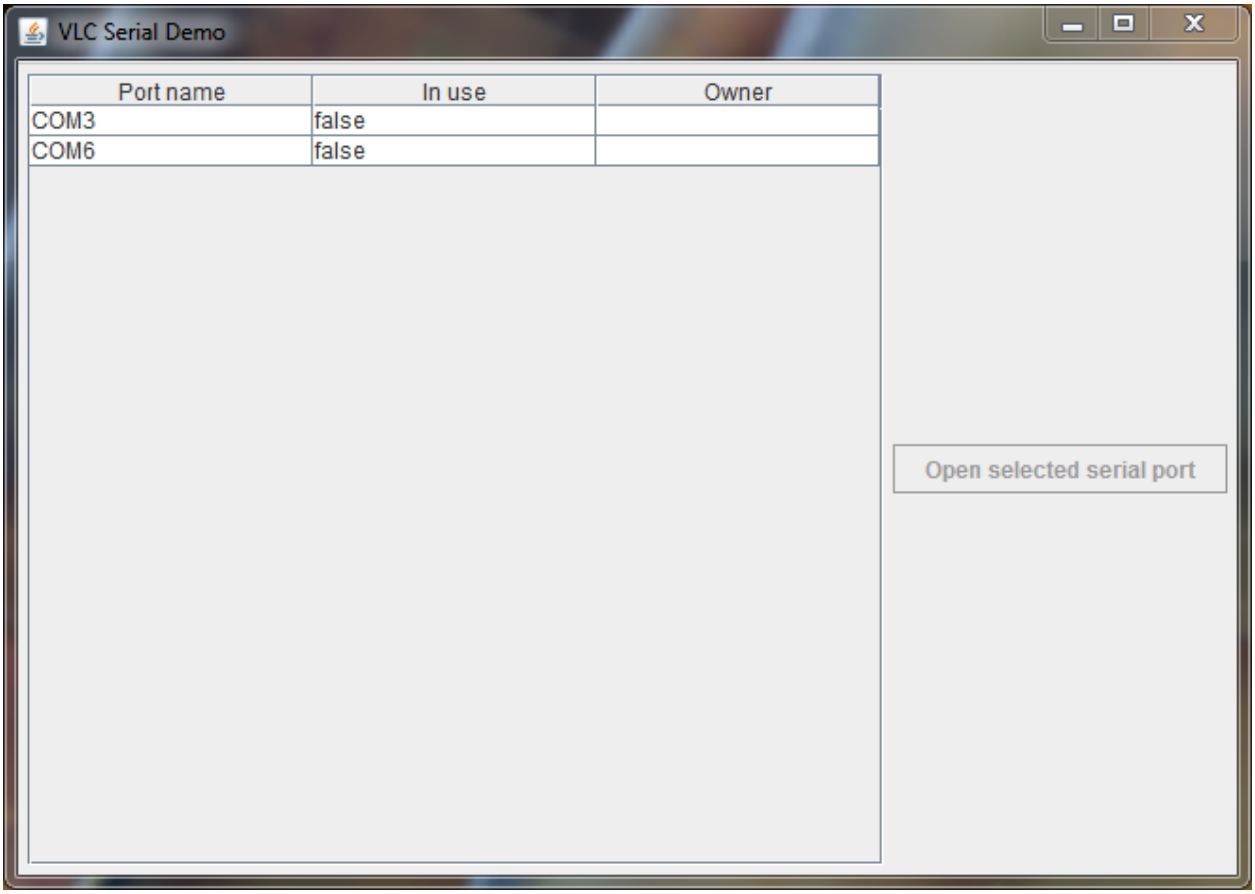

Figure 4: When the demonstration application starts, it prompts the user to pick a serial port to use. After a serial port is selected, the "Open selected serial port" button becomes enabled and the user may click the button it to continue.

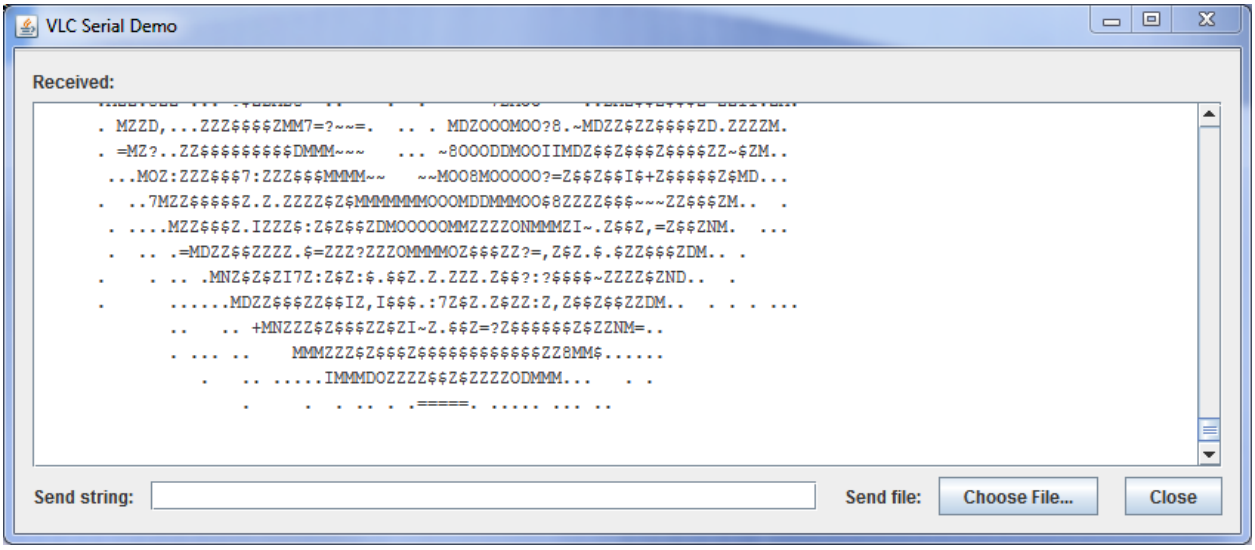

Figure 5: After opening the selected serial port, a serial terminal appears. The top text area shows the received data. A user may change what the serial terminal transmits by entering a new string into the "Send string:" text box or by choosing a file to send. The selected string or file is repeatedly transmitted until the serial port is closed. To close the serial port and exit the demonstration application, the "Close" button may be used.

5 Expected performance

The power supply and transceiver consume a total of 5W of power when the white LEDs are off and 22W when the LEDs are on.

The light output of this transceiver has not been measured yet.

The transceivers have been tested at 921600 baud, which is approximately 819.2kbps. At this speed, errors occurred frequently enough to be observable in a short period of time; the actual bit-error rate has not been measured yet. Most of these errors occur at the beginning or end of a message; this is probably due to the serial port leaving TXD high when there is no data to transmit[6].

The transceivers have also been tested with a 1MHz square wave[7] from a function generator. An oscilloscope at the receiving end was able to detect the square wave with some distortion and noise. This indicates that the transceivers may be able to support up to 2Mbps using NRZ encoding.

The transceivers have been tested at a 2 meter range. However, since the problem of aligning the transmitter with the receiver grows more difficult with increasing distance, the transceivers are typically used with a 1 meter separation.

6 Acronyms and abbreviations

DC direct current

FTDI Future Technology Devices International Ltd.

GND ground

GUI graphical user interface

[6]Since the transceiver was designed to handle Manchester code at 1Mbps, it is unable to keep the LEDs on continuously.

[7]This square wave alternated between 0V (which turns the LEDs off) and 5V (which turns the LEDs on) with a 50% duty-cycle.

JRE	Java Runtime Environment
kbps	kilobit per second
LED	light-emitting diode
Mbps	megabit per second
NRZ	non-return-to-zero
RXD	receive asynchronous data input
RX_OUT2	corrected receiver output
TXD	transmit asynchronous data output
TX_IN	transmitter input
USB	Universal Serial Bus
V	volt
VCP	Virtual COM port
VLC	visible light communication
W	watt