LR4

Laser Rangefinder Interface

February 5, 2013 Product Data Sheet

General Description

The LR4 is an interface board designed to add a USB interface to an off the shelf, hand held laser rangefinder. The LR4 works with a Fluke* 414D Laser Distance Meter. Together with the Fluke 414D, the LR4 provides ±3mm accuracy at distances up to 50 meters and sample rates up to 2.5 samples per second.

The LR4 implements a Human Interface Device (HID) class USB device. Software applications in most operating systems can easily read data from HID class devices without the need for custom drivers. This makes it simple to create software that interfaces to the LR4 from Windows or Linux.

* All trademarks are property of their respective owners.

Specifications

LR4 + Fluke 414D Combined Specifications

- Range: 50 meters
- Accuracy: ±3 mm
- Power Consumption: USB 5V at 180mA (laser on) / 25mA (laser off)
- Measurement Rate: 2.5Hz (good reflection strength), 500ms reacquire time when target distance changes significantly. Under poor reflectance conditions the measurement rate can drop to 0.5Hz.
- Operating Temperature: 0°C – 40°C
- Storage Temperature: -25°C – 70°C

LR4 Interface Board

- Size: 48mm x 46.5mm, fits inside Fluke 414D case (small cutout needed for USB cable)
- Weight of LR4: 9.1g (with supplied flex cables)

- Weight of Fluke 414D: 16.1g (removed from yellow outer case)
- Construction: 0.062” FR4
- Interfaces
    - USB mini “B” connector, female – to host
    - 8 pin FPC – connects in place of meter’s LCD
    - 6 pin FPC – connects in place of meter’s keypad
    - +3.0v power connection via solderless terminal block – in place of “+” battery connection
    - Ground connection via solderless terminal block – in place of meter’s “-“ battery connection
- Logical Host Interface: Generic Human Interface Device (HID)
- Standard keyboard emulation mode for direct data entry
- Measurement Modes: Continuous, Single Shot, Interval (configurable from 1 second to 1 year)
- Measurement Errors: Filtered or passed through modes

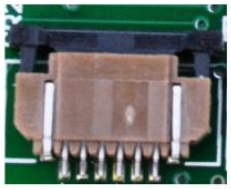
Connections

Please see our website (PorcupineElectronics.com) for a detailed video that shows how to install the LR4 inside the case of a Fluke 414D distance meter.

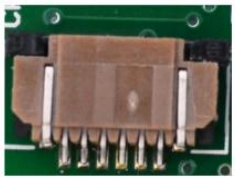
Note that care must be taken when installing the two flex cables as the conductors only make contact on one surface of the cable. Both flex cables should be installed with the contacts facing up (away from the surface of the circuit board). Also be sure to close the slide locks on the connectors after installing the flex cables.

Flex Cable Slide Lock

Open



Closed



Reference Software

The LR4 demo software (available on PorcupineElectronics.com) includes two Windows applications, LRDemo and LRSimple, that show how to create software that interfaces to the LR4. Full source code is included for both of these applications under a BSD style license. See the license information at the top of each source file for more information.

LRSimple

The LRSimple application is a bare bones Win32 Console app that implements the simplest method to start the LR4 and read distance measurements. If you intend to integrate the LR4 into a larger custom application, then this source code is the place to start.

LRDemo

The LRDemo application is a full featured Windows MFC application that demonstrates all of the features of the LR4. It is somewhat more complex than the LRSimple app and probably should be used as a reference when needing to implement more advance features.

Functional Description

The LR4 is a USB Human Interface Device (HID). Most modern operating systems have built in support for USB HID class devices. For example, Microsoft Windows allows application programs to communicate with HID class devices directly without the need to install device drivers.

USB HID class devices implement two pipes, one is an Out Endpoint and the other in an In Endpoint. The Out Endpoint is used to send commands to the LR2, and the In Endpoint receives status and measurement data. Both endpoints use 8-byte packets for all commands and status data. These commands and status packets are described in the sections below.

Sending Commands

All commands to the LR4 are 8-bytes in length. The first byte contains the command code, and the remaining seven bytes are used for any optional parameters required by the specific command. The command is sent to the LR4 using the writeData function in the ATUSBHID.DLL library (included with the sample software). The code snippet on the right shows how to send the GetConfig command.

```
UCHAR CmdBuf[8];
if (IsConnected) {  // If our device is attached

    CmdBuf[0] = CMD_GET_CONFIG;
    CmdBuf[1] = 0;
    CmdBuf[2] = 0;
    CmdBuf[3] = 0;
    CmdBuf[4] = 0;
    CmdBuf[5] = 0;
    CmdBuf[6] = 0;
    CmdBuf[7] = 0;

    DYNCALL(writeData)(&CmdBuf[0]);
    // In a few milliseconds the rangefinder will
    // respond with an Out report packet that will
    // be read by OnTimer function
}
```

Receiving Status Packets

```
UCHAR MeasurementBuf[8];
unsigned int Millimeters;
while (!_kbhit()) {
    if (DYNCALL(readData(&MeasurementBuf[0])) == TRUE) {
        if (MeasurementBuf[0] == STS_MEASUREMENT_DATA) {
            Millimeters = (MeasurementBuf[2] << 8) + MeasurementBuf[1];
            printf("%d mm\n", Millimeters);
        }
    }
}
```

All data is read from the LR4 in the form of 8-byte status packets. The first byte of each status packet indicates the type of status contained in the remaining bytes of the packet. The most common status packet type is a measurement data packet. Status packets are read from the LR4 using the readData function in the ATUSBHID.DLL library (included with the sample software). Normally data is read in a while loop or timer routine that runs several times per second. The code snippet to the right shows how to read and display measurements from the LR4. See the sample code for more details.

Measurement Status Packets

The LR4 always sends measurement data in the form of 8-byte status packets. The format of Measurement Data packets is shown below.

Response Packet Offset	Value	Description
0	0x00 (STS_MEASUREMENT_DATA)	Indicates packet contains measurement data
1	Measurement data (lsb)	Each measurement is a 16-bit unsigned integer with units of millimeters.
2	Measurement data (msb)	
3	Reserved	Reserved for diagnostic information.
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	

Note: The measurement packets sent by the LR4 are always in units of millimeters. The host side application may convert the measurement to any units needed before display or storage. This is done so that the measurement packet will always have the same format, and to avoid the complexity of encoding floating point data in a small packet.

Command: GetConfig (0x00)

The GetConfig command is used to read the LR4's configuration data from its internal non-volatile memory. The configuration data holds various settings such as the preferred display units, measurement interval time, etc.

Command Packet Offset	Value	Description
0	0x00 (CMD_GET_CONFIG)	Command code
1	0x00	Unused bytes (set to 0)
2	0x00	
3	0x00	
4	0x00	
5	0x00	
6	0x00	
7	0x00	

After receiving the GetConfig command, the LR4 will respond with an 8-byte packet containing its configurable values. The status packet typically takes a few milliseconds to transmit back to the host. The format of the Configuration Data packet is shown below.

Response Packet Offset	Value	Description
0	0x01 (STS_CONFIG_DATA)	Indicates packet contains configuration data
1	ConfigValue (lsb)	See below for details
2	ConfigValue (msb)	
3	Interval (lsb)	16-bit Measurement interval, units depend on ConfigValue
4	Interval (msb)	
5	0x00	Unused bytes (set to 0)
6	0x00	
7	0x00	

ConfigValue is a 16-bit structure defined as:

Bits	Field	Description
0:2	Distance Display 000 = Feet & Inches 001 = Meters 010 = Feet 011 = Inches 100 = Centimeters 101 - 111 = Reserved	Holds the preferred display units. This field is primarily used by the host to save the user's preference for displaying distance measurements. It is also used in keyboard emulation mode to "type" the measurement using the preferred units.

3:4	Measurement Mode 00 = Continuous 01 = Single 10 = Interval 11 = Reserved	Controls the operational mode of the LR4: Continuous: Performs and sends measurements as fast as possible. Single: Perform a one-shot measurement (see trigger methods below) Interval: Performs measurements at the rate specified in the Interval value.
7:8	Trigger 00 = Laser Rangefinder Run bit 01 = Caps Lock 10 = Num Lock 11 = Scroll Lock	Controls the event that triggers measurements to occur. In Single mode, each event triggers a single measurement. In Interval mode, the event toggles measurements on or off. This setting has no effect in Continuous mode.
9	Keyboard Emulation Mode 0 = Disabled 1 = Enabled	If enabled, the LR4 will simulate the measurement data being typed on a keyboard. This mode should only be used with Single or Interval mode to avoid flooding the system with keystrokes. See the section on keyboard emulation for more information.
10	Do Double Measurements 0 = Disabled 1 = Enabled	If enabled, the LR4 will perform two measurements and only send the result if the two measurements match exactly.
11	Don't Filter Errors 0 = Disabled (errors are filtered) 1 = Enabled (errors are returned)	During normal operation the LR4 occasionally reads erroneous data. Normally the LR4 filters these errors out. If enabled, this feature instructs the LR4 to return data from each measurement, even those that contain errors.
12	Only Send Changes 0 = Disabled 1 = Enabled	If enabled, the LR4 will only send a measurement packet if it contains a value different from the previous packet.
13:14	Reserved	Should be set to zero.
15	Laser Rangefinder Run 0 = Not running 1 = Running	If enabled, the LR4 will turn on the rangefinder's laser and start performing measurements.

Command: SetConfig (0x01)

The SetConfig command is used to write the LR4's configuration data. The SetConfig command changes the LR4's operating mode, but it does not store the parameters in the LR4's NVRam. The WriteConfig command should be used to save the LR4's configuration to NVRam. The configuration data holds various settings such as the preferred display units, measurement interval time, etc.

Command Packet Offset	Value	Description
0	0x01 (CMD_SET_CONFIG)	Command code
1	ConfigValue (lsb)	See the GetConfig section above for the definition of this field.
2	ConfigValue (msb)	
3	Interval (lsb)	16-bit Measurement interval, units depend on ConfigValue.
4	Interval (msb)	
5	0x00	Unused bytes (set to 0)
6	0x00	
7	0x00	

Command: WriteConfig (0x02)

The WriteConfig command is used to store the LR4's configuration data into its internal non-volatile memory. The configuration data holds various settings such as the preferred display units, measurement interval time, etc.

Command Packet Offset	Value	Description
0	0x02 (CMD_WRITE_CONFIG)	Command code
1	0x00	Unused bytes (set to 0)
2	0x00	
3	0x00	
4	0x00	
5	0x00	
6	0x00	
7	0x00	

Note: The LR4's configuration data is stored in flash memory. All flash memory degrades slightly each time it is erased and re-written. The LR4's internal flash can withstand 100,000 erase cycles. For this reason, application writers should make sure the WriteConfig command is not sent excessively.

Command: GetProductInfo (0x03)

The GetProductInfo command is used to read a six-byte block of the LR4's ProductInfo structure. This structure is made up of several ASCII strings that contain the LR4's manufacturer name, model, version, and serial number. The ProctInfo structure is defined later in this section. The entire ProductInfo structure can be read by repeatedly using this command to read 6-byte snippets of the structure.

Command Packet Offset	Value	Description
0	0x03 (CMD_GET_PRODUCT_INFO)	Command code
1	Offset	Byte offset into the ProductInfo structure.
2	0x00	Unused bytes (set to 0)
3	0x00	
4	0x00	
5	0x00	
6	0x00	
7	0x00	

After receiving the GetProductInfo command, the LR4 will respond with an 8-byte packet containing six bytes from the requested offset into the ProductInfo structure. The status packet typically takes a few milliseconds to transmit back to the host. The format of the ProductInfo packet is shown below.

Response Packet Offset	Value	Description
0	0x02 (STS_PRODUCT_INFO)	Indicates packet contains product info structure data.
1	Offset	Byte offset into the ProductInfo structure (same value as the offset requested in the GetProductInfo command).
2	Six bytes of data from the requested position within the ProductInfo structure.	See below for the definition of the ProductInfo structure.
3		
4		
5		
6		
7		

The ProductInfo structure is defined as:

```
typedef struct {
      char  ManufacturerName[30];
      char  ProductName[20];
      char  HardwareVersion[10];
      char  FirmwareVersion[10];
      char  SerialNumber[10];
} PRODUCT_INFO;
```

Note: Each field within the ProductInfo structure is an ASCII string. Each string is terminated by at least one zero character. Strings are left justified in each field and unused bytes are set to zero.

Keyboard Emulation Mode

The LR4 supports a keyboard emulation mode. When enabled, the host system will see key presses as if they were coming from a second USB keyboard. The LR4 will "type" out distance measurements using the units configured using the SetConfig command described above.

The LR4 is not aware of which program in the host system has input focus. Keystrokes are sent blindly. Care must be taken to only trigger measurements when the host computer has an appropriate application in focus, such as a spreadsheet or work processor. Other applications, such as the LR4 Demo App or the Windows File Explorer, will not be able to interpret the stream of seemingly random key presses. For this reason Keyboard Emulation mode should only be enabled in Single or Interval Mode with a trigger event enabled, Caps-Lock for example.

Here is an example sequence that demonstrates how to use Keyboard Emulation mode to automatically enter measurement data in Microsoft Excel:

1. Open Microsoft Excel.
2. Open the LR4 Demo Application (LRDemo.exe) downloaded from PorcupineElectronics.com.
3. Make sure the LR4 is attached to the host system.
4. Using the Demo App, set the Measurement Mode to "Interval."
5. Using the Demo App, set the Interval to 1 second.
6. Using the Demo App, set the Trigger to "Caps Lock."
7. Using the Demo App, enable Keyboard Emulation Mode.
8. Click on the cell within Excel that you wish to use to start receiving data.
9. Press the Caps-Lock key.

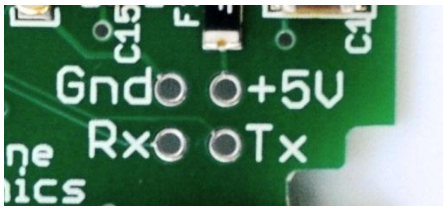
The LR4 will automatically perform a measurement once per second and enter that data into Excel. After each measurement the LR4 will simulate an "Enter" key press to move the cursor down one row.

10. Press the Caps-Lock key to stop the measurement data. Caps-Lock can be used to toggle measurement data on / off.

Serial Connection

The Porcupine Electronics LR4 interface board has hardware connections for a logic level serial port. The signals on this serial port are 3.0V "logic" level. Because of this, the LR4 cannot be directly connected to an RS232 port on a PC. The LR4 serial port is designed for interfacing directly to a microcontroller. If true RS232 signaling is needed, then a MAX3232 level shifter device must be used.

Serial port signals are available on a 2x2 header foot print on the LR4 board. The diagram below shows the signal positions as viewed from the component side of the board.



Signal Specification

Signal	Description
Gnd	Connect to power supply ground.
+5V	Power supply input for LR4 and the Fluke laser board. Power consumption is maximum of 200mA (laser on) / 30mA (laser off). Do not exceed 5.5V on this pin or permanent damage will result. If the LR4 is powered from this connection, it is not necessary to connect the LR4's USB port to a host system.
Rx	Input: LR4 UART receive data. Uses 3.0V signaling (3.3V tolerant). Logic low level: < 0.5V Logic high level: > 1.5V Do not exceed 3.3V on this pin or permanent damage will result.
Tx	Output: LR4 UART transmit data. Uses 3.0V signaling. Logic low level: < 0.5V Logic high level: > 2.3V This pin can source/sink up to 10mA.

Serial Protocol Specification

The Rx/Tx lines implement a standard logic level serial port without hardware flow control. The following serial communication parameters are used:

Parameter	Usage
Baud rate	9600 bps
Data bits	8
Parity	None
Stop bits	1

Serial Commands

The LR4 supports single byte commands sent from a host processor. No carriage return, line feed, or other delimiter character should be used. The following table documents the supported commands:

Command Name	Command String	Description
Go	g	The LR4 turns on the laser and begins measurement in continuous mode. After a few seconds, the LR4 starts transmitting measurement data as described in the measurement data section below.
Stop	s	The LR4 turns off the laser and stops sending measurement data.
Identify	i	The LR4 responds with identification and version information in the following format: Manufacturer: Porcupine Electronics Product: LR4 Hardware Version: Rev 2 Firmware Version: 1.1.0 Serial Number: 123456

Command Response

The LR4 responds to the Go and Stop commands with a short string indicating success or error. The table below documents the response strings supported by the LR4.

Response String	Meaning
ok	The LR4 interpreted the command and is executing the requested operation.
badcmd	An unsupported command was received.

Measurement Data

When the LR4 is performing measurements, it transmits measurement data over the serial link in ASCII format. Each measurement is a five digit numeric string representing the measured distance in millimeters. Each measurement is followed by a carriage return / line feed sequence. For example, the string “12345\r\n” means 12345 millimeters or 12.345 meters.

Legal Information

The Porcupine Electronics LR4 interface board is an after-market modification board designed to work with a Fluke 414D Laser Distance Meter. Neither the LR4, nor Porcupine Electronics are endorsed by Fluke Corporation. Removing the cover from a Fluke laser distance meter voids your warranty from Fluke Corporation. The LR4 board has been carefully designed with the intention of working with a Fluke 414D, but Porcupine Electronics cannot be held responsible for damage caused to a Fluke distance meter.

Disclaimer: The information in this document is provided in connection with Porcupine Electronics products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Porcupine Electronics products.

PORCUPINE ELECTRONICS ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL PORCUPINE ELECTRONICS BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF PORCUPINE ELECTRONICS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Porcupine Electronics makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Porcupine Electronics does not make any commitment to update the information contained herein.

Porcupine Electronics products are not suitable for, and shall not be used in, automotive applications. Porcupine Electronics products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

All trademarks are property of their respective owners.