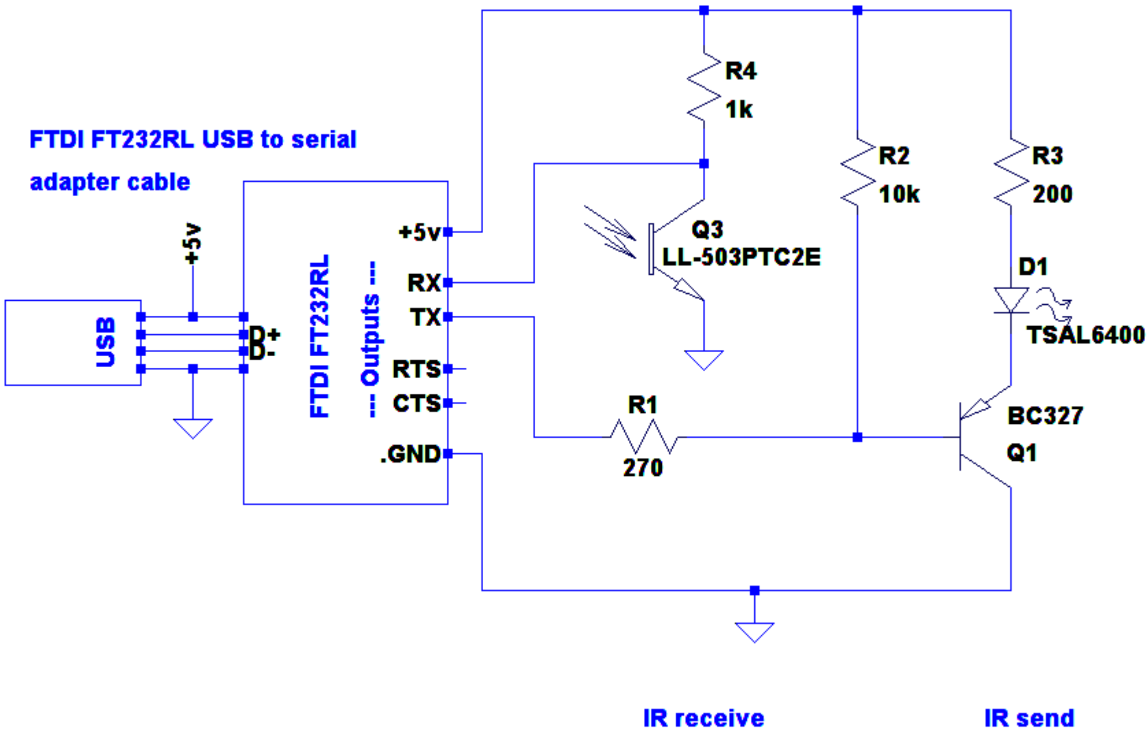
Fluke 187 IR Interface Using Discrete Components

IR Interface hardware

The original Fluke interface costs at least £45. If you are on a shoestring budget, a much cheaper interface can be constructed using an FTDI FT232RL cable and a few discrete components including the Infrared devices, a pair of NPN transistors and a few resistors.

The InfraRed devices were sourced from www.bitsbox.co.uk. The devices used were:

IR Transmitter:	Vishay TSAL6400 (https://www.bitsbox.co.uk/data/optos/tsal6400.pdf)
IR Receiver:	Vishay BPV10NF (https://www.bitsbox.co.uk/data/bp10nf.pdf)
Transistors:	BC337/2N2904 x 2
Resistors:	10k x 1, 1kΩ x 1/, 270Ω x 1, 200Ω x 1



The interface circuit

R3 limits the current through the IR led to below the maximum forward DC current. Peak current for the IR LED is 200mA. The value takes into account the 0.6v voltage drop of the switching transistor Q1. The IR LED will turn on when the voltage at the Q1 emitter is low which ocurrs when Q1 is turned on. The Tx line is high (3.5v) when idle and Q1 is turned when Tx goes low. Resistor R1 limits the current from the FT232RL transmit pin to no more than 20mA at 5v. During prototyping, a 270Ω resistor was actually used instead of the calculated value of 220Ω and this worked fine so was incorporated into the final build.

The IR LED can handle a maximum DC current 100mA and R3 was originally calculated to 33Ω for a current of just under that current. However, during testing, charcters were echoed back to the receiver, despite shielding. When the meter was moved away the echo stopped, suggesting IR output was intense enough to be reflected back from the meter IR sensor pannel. To reduce output intensity, R3 was increased and it was found that 200Ω seemed to work satisfactorily.

The receiver section simply consists of the IR receiver transistor with emitter connected to ground and collector to the receiver pin of the FT232RL and a 1k pull-up resistor R4 to the 5v rail.

Testing the interface

It is possible to test that the interface is working correctly by placing a desert spoon in front of the IR transmitter and receiver devices (which must be facing in the same direction). This reflects the transmitted signal back to the receiver creating a loopback mode. If the interface is working correctly, then characters typed at the terminal should then be echoed back and appear on the terminal screen as they are typed.

The terminal communication parameters should be set to 9600, N, 8, 1.

Connecting to Sigrok

Connecting the Fluke 187 to sigrok …

The interface is recognized as an FT232 USB-Serial (UART) adapter:

```
$ lsusb
Bus 002 Device 009: ID 0403:6001 Future Technology Devices International, Ltd
FT232 USB-Serial (UART) IC
...
```

The meter connection can be checked using a serial terminal, such as c-kermit or PuTTY. The terminal should be set up to communicate at 9600 baud, no partity, 8 data bits and 1 stop bit. Issuing the following command:

```
ID<CR>
```

The should result in the meter responding with its Make, Model, firmware version and serial number, e.g:

```
FLUKE 187, V2.02,0084700034
```

The meter should also respond to command such as QM<CR>:

```
QM,+30.548 mV AC
```

I have found the following commands do not work on the 187:

SF 19 – Backlight
SF 23 – Logging

SF 28 – Wakeup

Having confirmed that serial communication is working, we can proceed to working with sigrok. It is important to ensure that the IR devices are optically isolated from each other so that no stray IR light from the transmitter is picked up by the receiver. We do not want any sent characters to be echoed back otherwise sigrok will not repond correctly. The sigrok scan command responds as follows:

```
$ sigrok-cli --scan
The following devices were found:
demo - Demo device with 12 channels: D0 D1 D2 D3 D4 D5 D6 D7 A0 A1 A2 A3
ftdi-la - FTDI USB Serial Converter with 8 channels: TXD RXD RTS# CTS# DTR# DSR#
DCD# RI#
```

This confirms that the FTDI cable is recognized. To communicate with the meter we require the fluke-dmm driver, so we need to specify a command like:

```
$ sigrok-cli -d fluke-dmm:conn=/dev/ttyUSB0:serialcomm=9600/8n1 --scan
The following devices were found:
fluke-dmm - Fluke 187 V2.02 with 1 channel: P1
```

As shown, this should respond by identifying the meter and firmware version. We can also use:

```
$ sigrok-cli -d fluke-dmm:conn=/dev/ttyUSB0:serialcomm=9600/8n1 --show
Driver functions:
    Multimeter
Scan options:
    conn
    serialcomm
fluke-dmm - Fluke 187 V2.02 with 1 channel: P1
Supported configuration options:
    continuous: on, off
    limit_samples:
    limit_time:
```

This provides a bit more information. We can now try a command to read the meter:

```
$ sigrok-cli -d fluke-dmm:conn=/dev/ttyUSB0:serialcomm=9600/8n1 --samples 5
P1: 0.03 V AC RMS
P1: 0.03 V AC RMS
P1: 0.03 V AC RMS
P1: 0.03 V AC RMS
P1: 0.03 V AC RMS
```

In the event that the reponse is not as expected, change the debug level by adding `-l 4` or `-l 5`. This should return additional debug information which may help to track down the problem.

To use sigrok-meter, specify the command as follows:

```
$ ./sigrok-meter -d fluke-dmm:conn=/dev/ttyUSB0:serialcomm=9600/8n1
```

This should load a GUI environment. With the meter set to mV AC it should show a continuously updating random graph.

Interface notes

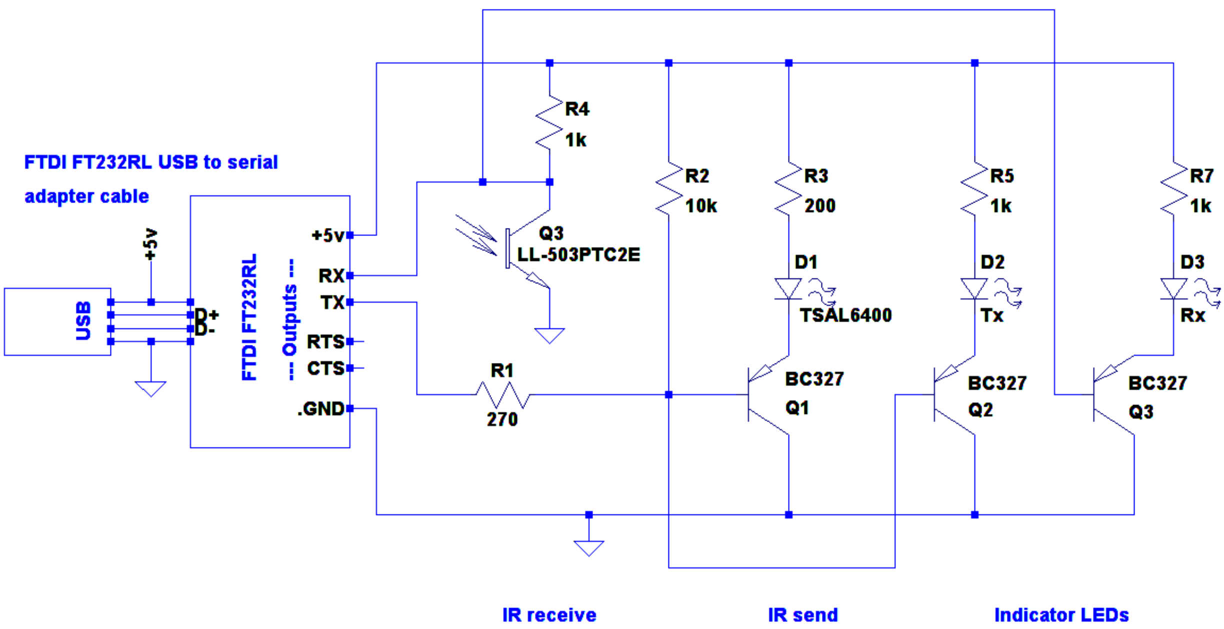
The interface uses an IR phototransistor. This should not be confused with an IR photodiode which is a different device but may look very similar. A IR photodiode is, in effect a photo sensitive diode, usually dark in colour that produces a voltage when IR light hits it. A IR phototransistor combines an IR photodiode with a transistor in the same package to provide an IR sensitive device that also has a gain characteristic. The IR diode is connected to the base of the transistor with the collector and emitter being exposed as pins. This should further not be confused with a 3 pin device used in 38kHz/40kHz applications where the third pin connects a frequency filter.

For interfacing with the Fluke 187, the serial communication parameters can be set to the usual default of 9600, N, 8, 1. The fluke-dmm driver used by Sigrok seems to also support a communication speed of 115,200 baud, however, there seems to be no way to switch the serial speed of the Fluke 187 from 9600 baud to 115,200 baud. This support may be included for other Fluke meters in the range (189/190 etc). The remote interface specification for the 189/187 indicates that communication parameters should be set to 9600 baud, no parity, 8bits, 1 stop bit.

The rise and fall time of the TSAL6400 IR LED seems to be too slow to operate at 115,200 baud, but works just fine at 9600 baud. Although fine for the 187/189, an IR interface for the 87v or 89v would need a faster device to run at higher speeds. There are other additional considerations for these meters.

Adding indicator LEDs

Adding indicator LEDs is easily accomplished using a couple of additional PNP transistors, a couple of LEDs and some additional resistors. The bases of the transitors are connected to the appropriate signaling points.



Circuit diagram of IR interface with indicator LEDs

For testing a red LED was used to indicate send and a green LED for receive, but any colour can be used. The 100Ω resistors are calculated for LEDs with 3.2v DC forward voltage and 20mA current. A

10mA LED would require a 180Ω resitor. The resistor should be caluclated to match the LED being used. If the brighness levels are too high, a higher value resistor can be used to reduce intensity and will draw less current. Anything up to 1k seems to work just fine.

General purpose PNP transistor types such as 2N2904, BC327 or BC177/8/9 should work just fine.