

# Using the USB Logic Analyzer with sigrok PulseView

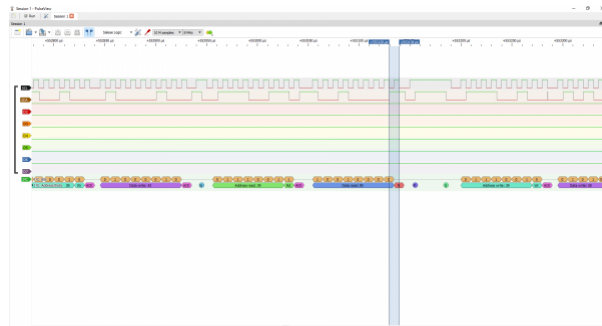
## Introduction

An 8-channel, 24MHz USB logic analyzer for under \$20. What a deal!



## USB Logic Analyzer - 25MHz/8-Channel ● TOL-15033

But a USB logic analyzer (LA) is only as useful as the software required to configure and monitor the tool. There are a few software options available for this USB logic analyzer; in this tutorial we aim to familiarize you with sigrok's PulseView.



sigrok is an open-source suite of software projects – all focused on supporting signal analysis tools. The project includes:

- PulseView – A logic analyzer front end with a simple GUI.
- sigrok-cli – A command line interface for sigrok – useful for scripting tests or running on a headless machine.
- fx2grok – A collection of open-source hardware LA layouts, schematics, and BOM's.

With an eye towards logic analyzers, this tutorial will focus mostly on PulseView.

## Get the Software

Download the latest PulseView release from sigrok's Downloads page. Here are direct links for the latest Windows, Mac, and Linux downloads:

- Windows
- Mac OS X
- Linux

Windows users can run the installer executable (**pulseview-NIGHTLY-32bit-static-release-installer.exe**) to install the software on your machine. The Mac installer is a binary disk image (**DMG**), which can be dragged into your Applications folder, for example.

**Heads up!** You may need to install drivers before using the USB logic analyzer. If you are using a Windows OS, check out the drivers section:

**WINDOWS DRIVERS FOR THE USB LOGIC ANALYZER**

Once installed open PulseView – the GUI frontend for sigrok.

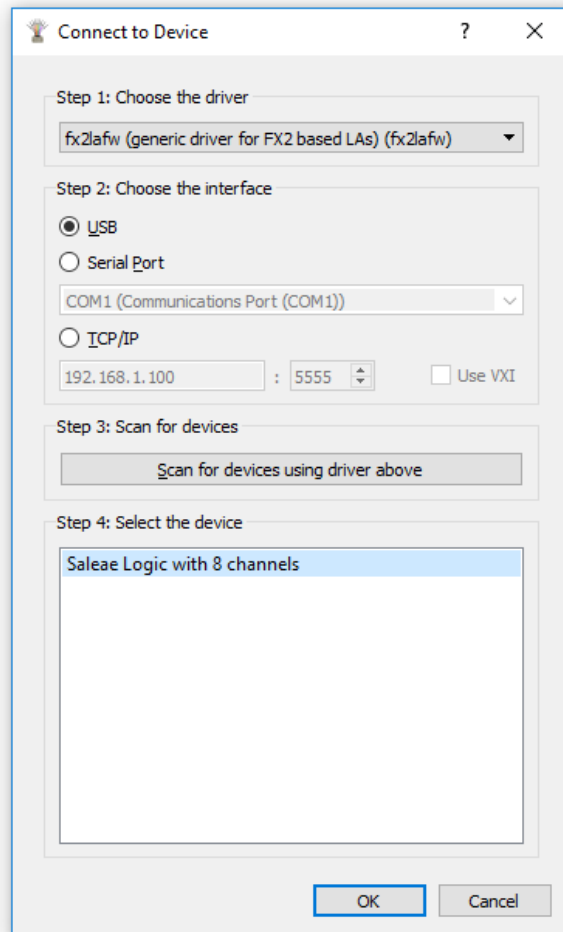
## Setting up the Software/Hardware

With PulseView open, plug in your USB Logic Analyzer. You should see faint red and green LEDs illuminate under the sticker.



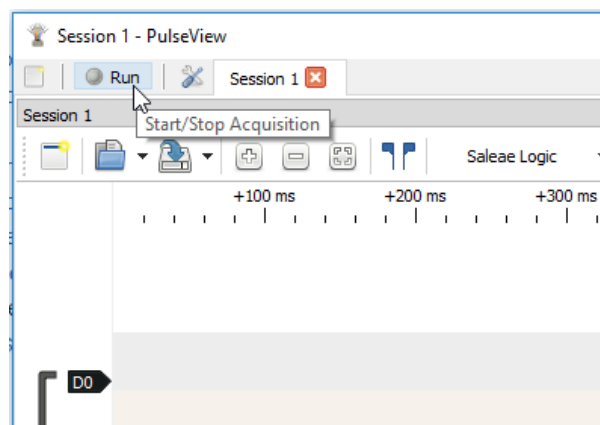
If PulseView does not automatically detect your logic analyzer, you'll need to set it manually:

1. Click the "<No Device>" dropdown menu.
2. Select **fx2lafw (generic driver for FX2 based LAs)** from the dropdown.
3. Select USB for the interface
4. Click **Scan for devices using driver above**
5. Select "Saleae Logic with 8 channels" and click "OK"



You'll be greeted with a blank slate of eight colorful bands of logic channels, numbered D0 to D7 (these match the CH0-CH7 labels on the LA).

Click the **Run** button in the top-left of the window to beginning scanning.



With the sampling parameters set as default – 1M samples, 20kHz – it'll take almost a minute to gather all million samples. You can hit "Stop" to end the scan early.

Unless you've already connected a few channels and grounds, this first scan will probably not be that interesting.

## Exploring the Capabilities

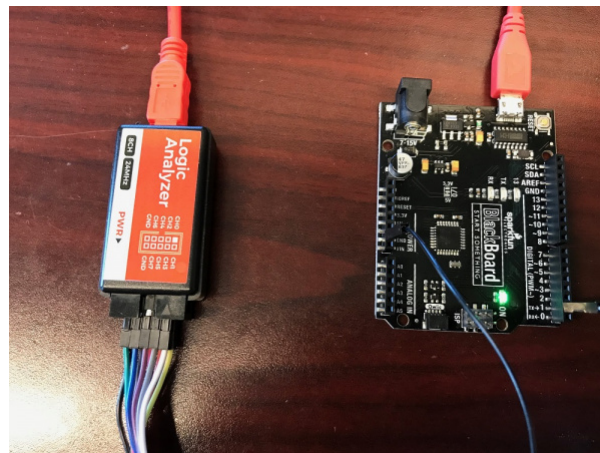
**Note:** This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE.

Here's a fun, tormenting Arduino sketch you can load to help familiarize yourself with PulseView's capabilities:

```
void setup() {
  // put your setup code here, to run once:
  randomSeed(analogRead(A0));
  Serial.begin(random(1, 115200)); // Set the baud rate to a random value between 1 and 115200 b
  ps
}

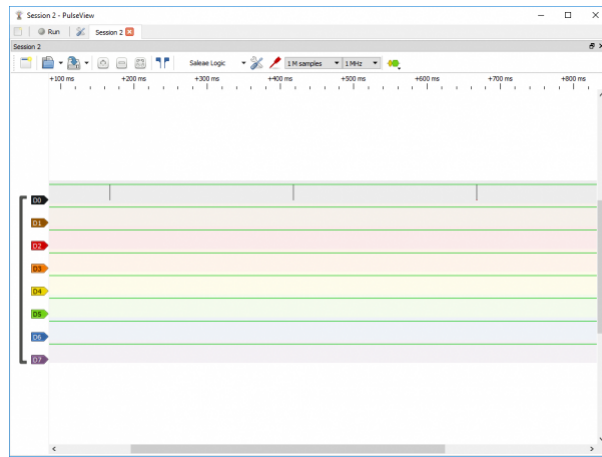
void loop() {
  Serial.println(millis()); // Print the time
  delay(250);
}
```

Load that onto an Arduino, then connect the logic analyzers "CH0" to your Arduino's TX pin (pin 1). Also connect one of the GND wires to GND.



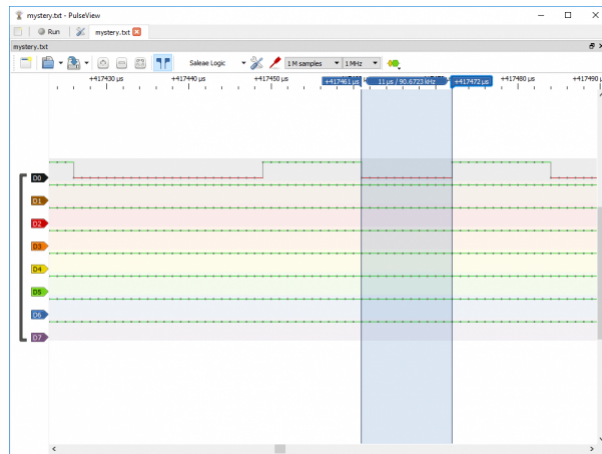
Before scanning, bump up the sample rate to **1MHz** and change the sample quantity to **1 M samples**. Depending on what you're trying to analyze, these dropdowns may get a lot of use. With those values set, hit **Run**.

You should be greeted with a seconds worth of samples, and a few blips on the D0 channel every 250ms.

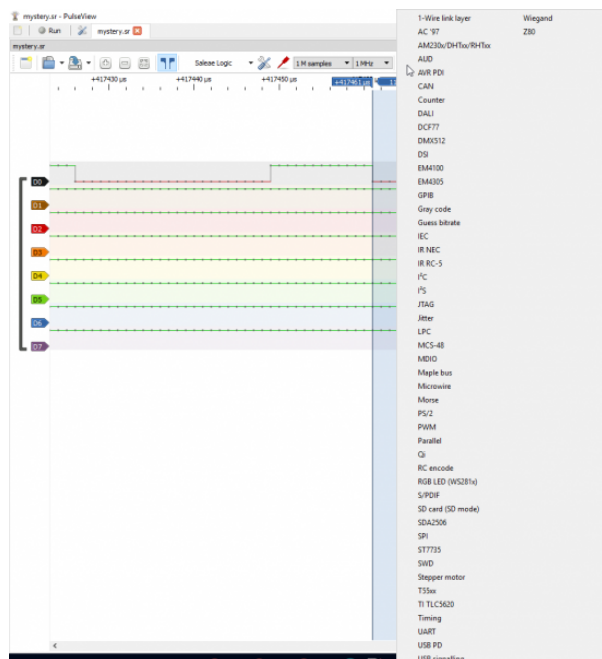


You can use your mouse's scroll wheel to zoom in and out, or use the "+" and "-" buttons on the toolbar. Zoom into one of the blips. Now it's your job to guess the baud rate!

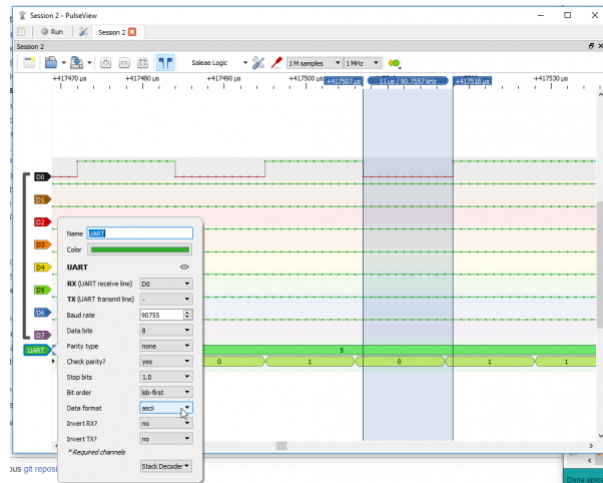
The **Show cursors** (pair of blue flags icon) tool can be useful for measuring time. Click that, then try to place the cursors around one bit of the transmission. The measured frequency should be our mystery baud rate!



To decode the string, use the **Add low-level, non-stacked protocol decoder** tool (looks like yellow and green decoded signals). Then select **UART** – note that a huge list of protocols pops up here, including I<sup>2</sup>C, I<sup>2</sup>S, SDIO, and SPI.



Click the green “UART” icon that appears towards the bottom-left and change the baud rate to your measured frequency. You can also change the data format to **ascii** to make the data easier to parse.



Now if you zoom out you should see your serial prints decoded!

## Resources and Going Further

For more information, check out the resources below:

- sigrok
  - PulseView
    - Windows
    - Mac OS X
    - Linux
  - sigrok-cli
  - fx2grok

### Further Explorations with the CLI

If you're connecting the logic analyzer to a headless machine, or want to automate a LA-based test, check out sigrok-cli – a command line interface for sigrok. With sigrok-cli installed, for example, you can use a command like:

```
sigrok-cli.exe -d fx2lafw --time 3000 --channels D0=RX --config samplerate=1m -P uart:baudrate=15200
```

To decode a UART connected to channel 0.

```
MINGW64/C:/Users/jacob/Desktop/Projects/LogicAnalyzer
uart-> 37
uart-> Stop bit
uart-> Start bit
uart-> 1
uart-> 1
uart-> 1
uart-> 0
uart-> 1
uart-> 1
uart-> 1
uart-> 0
uart-> 0
uart-> 37
uart-> Stop bit
uart-> Start bit
uart-> 1
uart-> 0
uart-> 0
uart-> 1
uart-> 1
uart-> 0
uart-> 0
uart-> 33
uart-> Stop bit
uart-> Start bit
uart-> 0
uart-> 1
uart-> 0
uart-> 1
uart-> 1
uart-> 0
uart-> 0
uart-> 34
uart-> Stop bit
uart-> Start bit
uart-> 1
uart-> 0
uart-> 1
uart-> 1
uart-> 1
uart-> 1
uart-> 0
uart-> 0
uart-> 35
uart-> Stop bit
uart-> Start bit
uart-> 1
uart-> 0
uart-> 1
uart-> 1
uart-> 0
uart-> 0
uart-> 00
uart-> Stop bit
uart-> Start bit
uart-> 0
uart-> 1
uart-> 0
uart-> 1
uart-> 0
uart-> 0
uart-> 0A
uart-> Stop bit
$ sigrok-cli.exe -d F421afw --time 5000 --channel0 DD4RX --config samplerate=1e --uart:baudrate=90909
```

*Click the image for a closer look.*

The CLI has a lot of potential for automation, and the main page is super-helpful!

As you venture into this world of logic analyzing, be sure to try out all of PuseView's protocol decoders and features. It's a great software tool and has a powerful open-source community behind it.