

USB 8-Channel Sensor Interface Linux Demo

Model 2218 | Rev.1.0.0 | February 2024

Copyright © 2024 Sensoray

SENSORAY | embedded electronics



Designed and manufactured in the U.S.A.

SENSORAY | p. 503.684.8005 | email: info@SENSORAY.com | www.SENSORAY.com

7313 SW Tech Center Drive | Portland, OR 97203

Table of Contents

Chapter 1: Introduction	1	1.1.2 Analog Capture	4
1.1 Demo Applications	1	1.1.3 Gpio settings and control	4
1.1.1 2218 Linux GUI demo	1		

Chapter 1: Introduction

1.1 Demo Applications

Included in the Linux SDK are 2 demo applications. There is a GUI application implemented in python with gtk, and a command line demo. This document describes usage of the Linux GUI demo. The other demo in Linux is a command line in C. The source code for both applications and middleware is included.

1.1.1 2218 Linux GUI demo

Start the Linux GUI application with the command “./app2218.py”. This python application uses the module s2218.py, which translates and maps API commands from python to the C API.

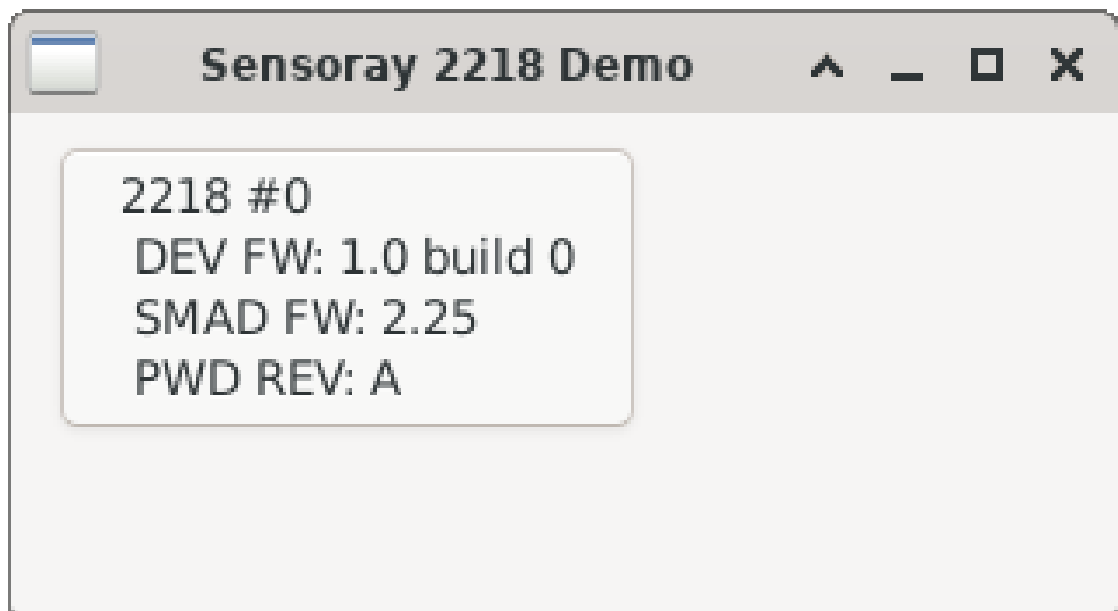


Figure 1: Model 2218 Python Demo One Device Attached

When the demo first starts, it scans continuously for any attached 2218s. Figure 1 shows the results for one attached 2218. If no 2218 boards are present, a dialog showing the scan status will be displayed (Figure 2). It detects any 2218s that are plugged in by continuously polling s2218_OpenApi in the API.

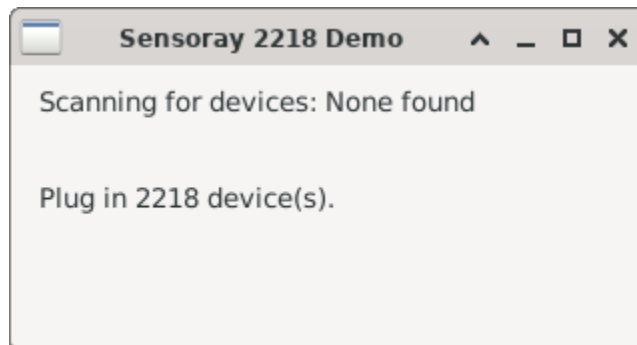


Figure 2: Model 2218 python demo with no devices attached

Clicking on the desired device (Figure 1) will open the Analog dialog (Figure 3). The version of the device and firmware will be displayed at the top of the dialog. By default, each sensor channel is in the disabled state. The device will have its analog circuitry powered down when it is first plugged in. Click on 15V and -12V to enable the device's analog power rails. If only using digital IO, you may want to keep the analog power disabled to conserve power. Clicking on status with the power levels disabled will display a valid digital voltage (DIG=3.3V), but invalid analog voltages (Figure 3).

After enabling 15V and -12V, clicking status should result in valid analog voltages being read back (Figure 4). It will also display the current device temperature in degrees Celcius.

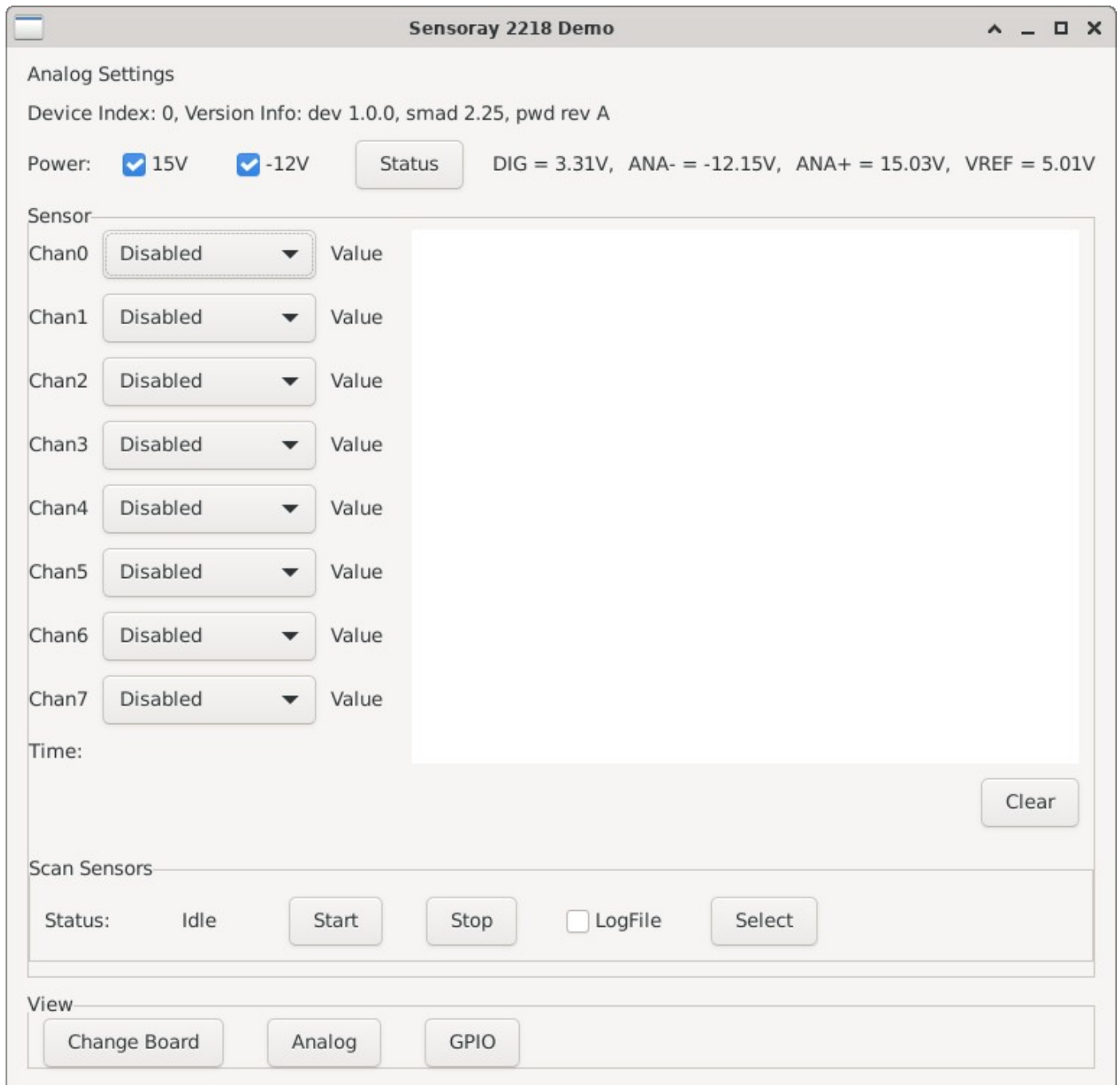


Figure 4: Analog Settings and Control Dialog (Analog Power Enabled)

1.1.2 Analog Capture

To capture analog data, first set up any capture channels of interest to the desired sensor type. Start scanning (reading analog values) by clicking the “Start” button under scan sensors. There is also an option to log the raw analog data to file. The capture data is comma delimited with the device time as the first field, followed by the sensor values. Please see the technical resource manual for further details on capture values and how to connect the analog channels.

1.1.3 Gpio Settings and Control

Switch to the GPIO settings by clicking the option in the “View” panel. The GPIO dialog will appear similar to Figure 5.

The screenshot shows the 'Sensoray 2218 Demo' application window. At the top, it displays 'GPIO Settings' and 'Device Index: 0, Version Info: dev 1.0.0, smad 2.25, pwd rev A'. Below this is a table for configuring 8 GPIO pins (GPIO 0 to GPIO 7). Each pin has a 'Direction' dropdown (all set to 'Input'), a 'Debounce(0-255ms)' section with 'Rising' and 'Falling' edge triggers (all set to 0), and a 'Pull-up, Pull-down bias' section with 'bias' (all 'Off'), 'polarity' (all 'Down'), and 'value' (all 1) dropdowns. Below the table is a 'Continuous Read(1s refresh)' section with a 'Status' label (set to 'Idle'), 'Start', 'Stop', 'LogFile' checkbox, and 'Select' buttons. A 'Refresh' button is located to the right of this section. At the bottom, a 'View' panel contains 'Change Board', 'Analog', and 'GPIO' buttons.

Figure 6: GPIO settings and control

Each GPIO channel may be selected as either an input or an output. The input bias can be enabled to pull down or pull up each individual input. Debouncing is also configurable from 0-255ms on the rising or falling edge.

The “Refresh” button queries the current state of all GPIOs and displays the read inputs in the Value column on the right side of the dialog. Additionally, the GPIOs may be read continuously by pressing the “Start” button under “Continuous Read”. If desired, the GPIO data may be written to file with the LogFile option. The polling rate for the continuous read scan is once per second. The rate may be changed in the python source code file `app2218.py`.

1.1.4 Changing Boards

The selected board can be changed by either restarting the demo program or clicking “Change Board” in the “View” panel at the bottom of the demo. The board selection dialog scans for new boards and will detect any boards plugged in or removed from USB.