Installation Instructions

APSpec2148 driver python3 installation instructions.

APSpec interacts with PC through USB bus. To work with APSpectrometers in python/Ubuntu environment, you may install opensource pyftdi user space driver for popular FTDI devices. Please go through the installation section of pyftdi driver and install pyftdi package.

Alternatively you may write your own C/C++ application code and interact with Aipplica Spectrometers via FTDI's D2XX API drivers (Windows) or libftd2xx (Linux) & FTDI's MPSSE library and Aipplica Spectrometer library, libAPSpec. Go to the next section for instructions.

For Python/Linux Environment:

$pip3 install pyftdi

Please note, as outlined in the pyftdi installation section, On Linux, you also need to create a *udev* configuration file to allow user-space processes to access to the FTDI devices. After you have successfully installed pyftdi.

Download APSpec2148 latest APSpec_*.whl from www.aipplica.com/downloads.html

$pip3 install APSpec-0.1-py3-none-any.whl

Now you are ready to use the APSpectrometer. For plotting and viewing captured spectral data, these scripts use matplotlib. Please ensure matplotlib is installed.

$pip3 install matplotlib

Invoke Python3 and follow the sequence of commands as provided in **DemoA.py** or if you use jupyter notebook, invoke jupyter notebook with **DemoA.ipynb** These files are provided in Download section on Aipplica website.

For DemoA.py executable file you may invoke the python3 executable as follows,

$./DemoA.py

but first read through the comments to understand that you need to block the optical input when collecting the dark signal and illuminate the slit/fiber when collecting the light spectra at the right steps. The scipt will open the plot and display the captured signal. It will proceed to the next step after you close the graphics plot. Once you understand the script, you may customize/modify as needed.

For DemoA.ipynb you may execute in jupyter notebook, Please read through the comments to understand the flow.

$jupyter notebook DemoA.ipynb

Installation Instructions

By this point you have instantiated a spectrometer module and captured the dark signal data. You may turn on the light source of your choice (Colour LEDs,LED torch light,light bulb and capture its spectra a plot it and process the captured spectrum as you please.

For C/C++ & Linux Environment

FTDI driver installation instructions.

APSpec uses FTDIs USB interface IC with Multi Protocol Synchronous Serial Engine (MPSSE) hardware. The MPSSE is a hardware block found in several FTDI chips which communicates with a PC over the USB interface. Applications on a PC communicate with the MPSSE in these chips using the D2XX USB drivers and the LibMPSSE library and the libAPSpec wrapper library.

Please install FTDI's libftd2xx drivers in Linux (D2XX API drivers for Windows) Please refer to FTDIs website for additional details. The following Application Note on FTDIs website may be helpful: *AN_220_FTDI_Drivers_Installation_Guide_for_Linux-1.pdf*

Next Download the LibMPSSE.zip library from FTDI's website: You will need the libMPSSE.h and libMPSSE.a files from here.

https://ftdichip.com/software-examples/mpsse-projects/libmpsse-spi-examples/

Download "APSpec C Library Programmer's Guide" and libAPSpectra.a & APSpectra.h from Aipplica website Download sections and compile with the libraries as follows

```
>gcc MyApp.c -o MyApp.o  ./libAPSpec.a ./libMPSSE.a ./libftd2xx.a -lpthread -ldl -lrt -Wall -
Wextra
OR
>g++ MyApp.cpp -o MyApp.o  ./libAPSpec.a ./libMPSSE.a ./libftd2xx.a -lpthread -ldl -lrt -Wall -
Wextra
```

(Temporarily remove ftdi_sio & usbserial drivers as they are not compatible with D2XX. You can restore those later with [sudo modeprobe ftdi_sio; sudo modeprobe usbserial] commands)

```
>sudo rmmod ftdi_sio; sudo rmmod usbserial
```

Plug in the APSpectrometer module and Run MyApp.o

```
>chmod +x MyApp.o
>./MyApp.o
```

if you invoked AP_InitUSB() in MyApp.c, You will see an output similar to

```
Description=APS2148        or APS<model number>
SerialNumber=160921006    or a <number string>
```