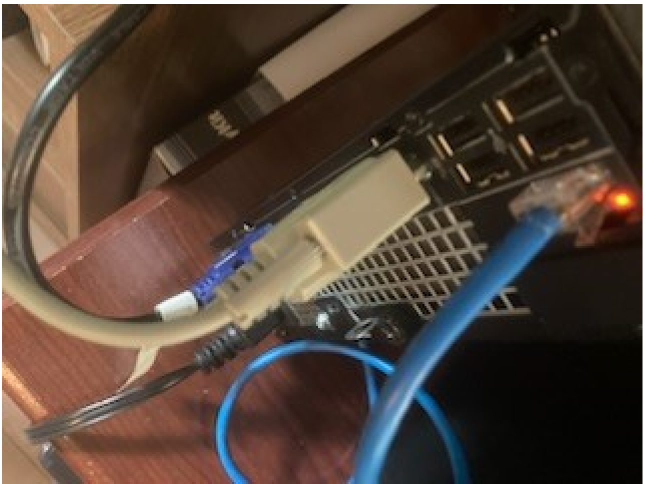
Radio Shack 22-812 data logging via RS-232 port

> *"If you want to monitor a measured electrical parameter (voltage, current, resistance, etc.) over a period of time, this python program can talk to a Radio Shack 22-812 digital multimeter over the serial port and print out its readings."* - from original posting (https://code.google.com/archive/p/rs22812/downloads)

In 2023 I still agree with the statement above made by Don Peterson the original creator. I give him credit for doing the heavy lifting. The format of the data is not exactly pleasant as Radio Shack's developers chose to output the segments that are lit on the display along with the mode information for the serial data. Why, who knows, strange formats just happen sometimes. I have worked in industry and have used SPI (Serial Peripheral Interface) and SCI (Serial Communications Interface) on industrial equipment and personally would not have done that. I would have sent the data in the format that Don has decoded it into. Basically he made an API to make the formatting of the data proper. But, here we have it, a meter from many years ago that works great, (not quite like a Fluke scopemeter!) but, good enough for most things. The only negative besides the strange data format is the fuses. The current measurement fuses especially the 500mA one are rather wimpy and a small mistake and it's blown. One that could handle a few Amps of a bench power supply before blowing would have been nice. These meters are still available on eBay at a reasonable price point. In 2023, they are available for less than 50USD. Documentation (https://www.google.com/search?client=firefox-b-1-lm&q=radio+shack+multimeter+22-812+manual) is still out there for it too and the CD files from the CD that came with it on Archive (https://archive.org/details/radioshackmeterview10_220812).



Serial ports are harder to find on computers in 2023. Luckily I have a Dell Optiplex 7010 which is a few years older and still has the port. I have noticed that many of these small form format units do have aerial port. I suspect this is due to the fact that they might be used to interface with cash registers that have legacy hardware requirements. So it might be possible to still get built in serial ports for a while

until all the legacy stuff that PCs can attach to gets recycled. The 7010 was bought primarily for use as a small media computer, replacing the function of a DVD player and so much more with a unit that fits where a DVD player does. It is quiet too with a laptop style cooling system. Quiet is good for a media PC as I have learned the hard way with the previous unit.
Serial port to USB converters are readily available on eBay as one might expect. So there is no real need to obtain a computer with a serial port these days.

The Code and Mods to it

The Google code repo is for Python < v3. I have updated the code so that it will run under Python > v3.

File names

I modified the file rs22812_linux.py as I am using Linux. In the archive I renamed it to rs22812_linux_legacy.py for clarity. I also archived rs22812.py, meter.py and Don’s original readme.pdf (worth a read to get a better understanding) for completeness and ease if anyone want to see it all in one place.

Updates to code

As the original post on the code points out there are some changes that need to be made in order to make the scripts run on newer Python code.

There is a note to this effect on the post…

> *“Update 17 Apr 2012: A user named J. Muczynski mentioned in an email that the following changes were needed for use with python 3.1:*
>
> - *add () for the print statements*
> - *remove ord() calls because the code is using integers, not characters*
> - *use PySerial version 2.6 when using Python 3.1*

”

I followed line of modification and as serial from Pyserial had to do the change related to setDTR.

1. Made print calls Python 3 compliant.
2. setDTR (used as self.sp.setDTR) depreciated, using self.sp.dtr = False
style instead.
3. Removed all ord() calls.

Testing

This updated version that works fine for me, was tested with...

```
Python 3.8.10 (default, May 26 2023, 14:05:08)
```

...and...

```
pyserial==3.5
```

This code is rs22812_linux_py3.py

More Mods – Logging Version

From the new Python 3 base I added logging to a file by default. Additionally I chose to change the date formatting, looking ahead to any parsing that I might want to perform.

rs22812_linux_py3_l.py

This code has the additional option for -l for logging to a specific file. It will always log though to output.log.
The logging output is identical to what is sent to stdout.

```
erick@ThinkPad:~/7010/python/meter$ tail output.log
05-Jul-2023,13:08:53 [571] ('22.8 kohm', 'ohm', ())
05-Jul-2023,13:23:53 [572] ('21.9 kohm', 'ohm', ())
05-Jul-2023,13:38:54 [573] ('19.5 kohm', 'ohm', ())
05-Jul-2023,13:53:55 [574] ('20.2 kohm', 'ohm', ())
05-Jul-2023,14:08:55 [575] ('21.8 kohm', 'ohm', ())
05-Jul-2023,14:23:57 [576] ('22.4 kohm', 'ohm', ())
05-Jul-2023,14:38:57 [577] ('22.5 kohm', 'ohm', ())
05-Jul-2023,14:53:58 [578] ('23.2 kohm', 'ohm', ())
05-Jul-2023,15:08:59 [579] ('23.7 kohm', 'ohm', ())
05-Jul-2023,15:23:59 [580] ('27.0 kohm', 'ohm', ())
```

Note the date format and comma between date and time.

Help Menu Expanded for Logging Version

```
Help for Command Line Options
----------------------------

Usage: rs22812_linux_py3_l.py [options]

rs22812 - A python interface for the Radio Shack 22-812 digital multimeter.

Options:
  -h, --help            show this help message and exit
  -p PORT, --port=PORT  port device string: examples:  Unix-like: /dev/ttyS0,
                        Windows: COM1 [defaults to one of these values]
  -i INTERVAL, --interval=INTERVAL
                        interval in seconds between readings [default: 1]
  -l LOG, --log=LOG     Log to a file. Takes filename as paramter. (default
                        output.log)
```

Additional Code

During the process of debugging, I came to a point where I had code that ran without errors but sat there and did nothing at all. I had yet to get the setDTR correct. I tried a few hacks on it that did not work before finding the PySerial documentation. I had dead air, nothing. But, I knew there was comm from the meter by using minicom to see what was coming in from the serial port.

I used ChatGPT to create a simple port scanner for COM ports. I was looking for something that works like netscan on Linux, just check the ports and report back what's ready to go.
This is the port-test.py file.

Also I have code that takes the output.log and plots it using Python's Matplotlib.

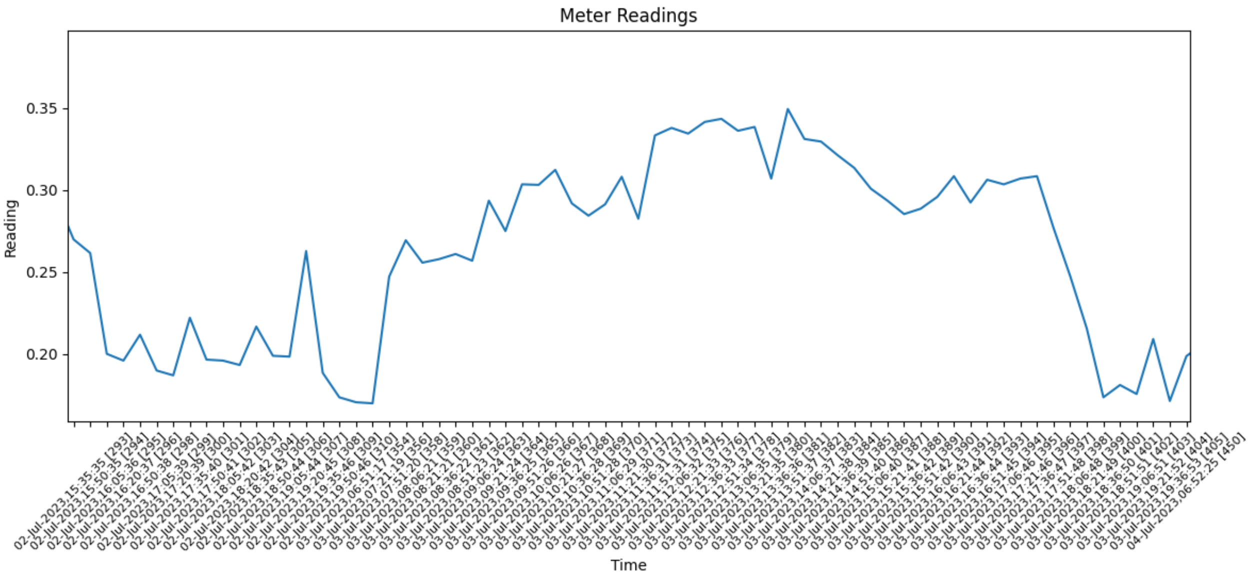
plot-meter-output-log-recip.py plot-meter-output.py

plot-meter-output.py – Just does regular plotting of the data.
plot-meter-output-log-recip.py – Is a quick hack of the above code to allow plotting of the 1/ln(data). I am taking to log and then the reciprocal. Why? Because what resurrected my interest in collecting data via the RS-232 line was that I was using a light sensor. The sensor has less resistance with more light input and I wanted it to show values in reverse, more light = higher numbers. Taking the log makes the formatting a bit easier to read on the plot as there is a lot of range between bright which could be 20k Ohms to dark which runs into the Megohms.
The plotting code will ignore OF (Overflows). It will not show them on the plot in other words. It is best to take the meter out of auto ranging and make it stay in one range or else the plot will not reflect reality as it will jump into various ranges if the data has high dynamic range, like the light sensor does. Using the log function to compress the range helps the plot.

The date, time and count appear on the X axis, diagonally. This gets a bit cluttered if the sample rate is high. In my case I was sampling ambient outdoor light every 15 minutes for testing the setup.

Note: I tried using the meter with these 9V batteries that is rechargeable via USB. A clever idea and the battery seems to last days in the meter on a single charge. I don't think I will be buying regular 9V alkaline batteries again. These rechargeable 9V batteries can be found on eBay and the price in 2023 at 4.50USD is reasonable.

Light Sensor Readings in 1/ln(data) format



References

I did try zmeter as well but, for me it did not produce output, it sat there without showing anything.

https://zmeter.sourceforge.net/
https://code.google.com/archive/p/rs22812/downloads
https://forum.allaboutcircuits.com/threads/inexpensive-datalogging-via-serial-port-and-radio-shack-multimeter.26744/