

A Sky Quality Meter Display

Peter D. Hiscocks

Royal Astronomical Society, Toronto Centre, Canada

Syscomp Electronic Design Limited

Email: phiscock@ee.ryerson.ca

Web: <http://www.ee.ryerson.ca/~phiscock>

September 8, 2010

Introduction

As part of a research program on light pollution, I needed to acquire a series of *dark sky quality* readings and tag each one with its location¹. The software needed to run on a Linux notebook. Unihedron [1], who manufacture the SQM-LU dark sky meter (figure 1(a)), supply the instrument with a display program that runs only under the Windows operating system. Fortunately, the SQM-LU manual documents the instrument commands in detail, and the instrument uses the FTDI USB-Serial technique. This means the device can be addressed via a serial port, making the Linux software relatively straightforward.

The program described here (figure 1(b)) runs without changes on a Linux or Windows machine. With a small amount of additional development, it would also run on a Mac under the OS-X operating system².

The program is written in the Tcl/Tk language. A quick introduction to the Tcl/Tk language is in [2].

The Tcl language is interpreted (like Java), so it is straightforward to install and execute. However, it is orders of magnitude easier to program in Tcl.

The Graphical User Interface

- **Magnitude per Square ArcSecond** Primary output of the instrument, a measure of the darkness of the sky. Larger number is darker.
- **NELM, Naked Eye Limiting Magnitude** An estimate of the magnitude of the dimmest stars visible to the naked eye, under these light conditions.
- **Temperature** The SQM light detector is temperature compensated. Temperature reading is available as a result.
- **Date and Time** This display shows a *time stamp* used in the data records. It's also a useful indicator that the display is working correctly (the 'seconds' indicator changes).



(a) Instrument



(b) Linux Display

Figure 1: Sky Quality Meter and Linux Display

¹The results of that project will be reported in a separate paper.

²The Tcl source code is available on the author's website, at <http://www.ee.ryerson.ca/~phiscock>

- **Manual Update Button** The yellow 'manual update' button causes the instrument to request a new reading from the hardware. It may be actuated even when the instrument is auto-updating.
- **Manual/Auto Update** These are a radio-button pair: actuating one deselects the other. In auto-update mode, the instrument updates its reading at the time interval specified in the **Interval, Sec** entry widget window.
- **Recording On/Off** Another radio-button pair, controlling the recording of data to a disk file. Left-clicking on **Recording On** calls up a file selector menu that allows one to specify the name of the data file. Data is stored in csv (comma-separated values) format, suitable for loading into a spreadsheet.
- **File Records** This shows the number of records written to a file. It's useful to monitor recording, to ensure it's taking place correctly.
- **Port Selector** This is a spinbox which allows one to select the appropriate serial port. The list automatically adjusts to Windows or Linux port names.
- **Port Status** This indicator shows **Connected** or **Disconnected** for the status of the serial port connection.
- **Connect** This button allows one to connect, after selecting the correct serial port.
- **Console Show/Hide** Under linux, the GUI is automatically accompanied by a *console* which show status and error messages. Under Windows, it's possible to show or hide the console, as controlled by these radiobuttons. These buttons are not shown on the Linux GUI.

A typical console window display is shown in figure 2. It is particularly useful for displaying debugging messages during the process of software development.

Verifying the Port Connection

It's advisable as a first step to make sure the computer is recognizing the SQM hardware as a serial-port device. Here's how you do that on Windows and Linux machines.

Windows: Checking the COM port Connection

Right click on My Computer

Navigate to My Computer -> Manage -> Device Manager

Double click on Ports (COM and LPT)

You should see something like USB-SQM Port (COM3) If you want to change this port assignment, then right click on that entry and select Properties

Navigate to Port Settings -> Advanced and select the desired port number.

If you know that this COM port is available, you can ignore any warning messages about other equipment using this COM port.

Now return to the SQM display. Select that same port in the **Port Selector** window, and press **Connect**. After a brief pause, the status should change from **Disconnected** to **Connected**.

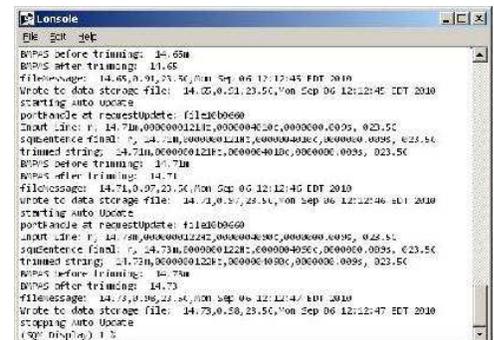


Figure 2: Console Window

Linux: Checking the COM port Connection

On a Linux machine, the USB-Serial ports are assigned on a first-come first-served basis. To determine the assigned port, plug in the hardware to a USB port, open a terminal window and run the program `dmesg`. At the end of the message, you should see something like this:

```
usb 3-1: new full speed USB device using uhci_hcd and address 7
usb 3-1: new device found, idVendor=067b, idProduct=aaa0
usb 3-1: new device strings: Mfr=1, Product=2, SerialNumber=3
usb 3-1: Product: USB-Serial Controller D
usb 3-1: Manufacturer: Prolific Technology Inc.
usb 3-1: SerialNumber: 12345678
usb 3-1: configuration #1 chosen from 1 choice
pl2303 3-1:1.0: pl2303 converter detected
usb 3-1: pl2303 converter now attached to ttyUSB2
```

This indicates that the hardware unit is attached to `/dev/ttyUSB2`, Select that port in the **Port Selector** window, and press **Connect**. After a brief pause, the status should change from **Disconnected** to **Connected**.

Running the Program

Windows

If the Tcl language is not installed on your Windows machine, download it from the ActiveState website: <http://www.activestate.com/activetcl> (use the Free Download button).

Plug in the SQM hardware and check device manager (under Ports:COM and LPT) to determine the assigned COM port. You can use the COM port properties to change the assignment some other port number.

Once the ActiveState Tcl interpreter is downloaded, any file ending in `.tcl` should be associated with the Tcl interpreter. Then you can start the program by double-clicking on the source code icon for `sqm-display.tcl`. Alternatively, start the Tcl/Tk interpreter by double-clicking on the `wish` icon. A terminal window will appear. In that window, type `source sqm-display.tcl`. The program should start.

Select the COM port and then click on **Connect**. The Status window should show **Connected** and you should be able to do a manual update of the display.

Linux

Verify that the Tcl interpreter is present by typing into a terminal window the command `which wish`. You should get an answer like `/usr/bin/wish`, indicating that Tcl is present.

Then you type into a terminal window the command `wish sqm-display.tcl` which will start the program.

Alternatively, you can start a Tcl interactive window with the command `wish`. Then, in that window, type `source sqm-display.tcl` to start the program.

When the GUI appears, select the serial port (`/dev/ttyUSB0` or whatever) and then click on **Connect**. The Status window should show **Connected** and you should be able to do a manual update of the display.

Mac

The code has not been tested on a Mac but with a small amount of additional development (setting the port names correctly) it should run.

If the Tcl language is not installed on your machine, download it from the ActiveState website: <http://www.activestate.com/activetcl> (use the Free Download button).

Ensure the Tcl interpreter is downloaded and installed. Plug in the hardware. Determine the port that the hardware is assigned to, which might be something like `/dev/cua0`. Using the Linux and Windows port assignment code as a guide, modify the source code to add detection of the port on a Mac.

Details of the Software

Why Use Tcl/Tk for this Project?

The Tcl/Tk language is ideal for a project of this sort:

- Tcl/Tk is available as Open Source, that is, it is freely downloadable and the source code is available. It also runs under Windows, Linux and Mac operating systems, with very few – if any – differences³.
- The code is interpreted⁴, so there is no explicit edit-compile-debug cycle. It's shortened to an edit-run cycle. Debug information surfaces during the run of the code.
- It's trivial to add print statements to the code to monitor the behaviour of variables and determine the flow of control.
- It is very simple to construct a graphical user interface, and relatively few instructions can do the job.
- The Tcl language can be operated in command line mode (direct entry of commands) or by interpretation of source code in a text file. Command line mode allows one to very quickly test the operation of a command or group of commands.
- The Tcl language is a *scripting* language, and as such it has a wide range of string manipulation commands.
- An *event loop* is automatically created for handling of button events and incoming messages from the serial port. All of the machinery to implement this is hidden from the user.
- The language is very well supported with numerous books and a Tcl WIKI.

Overall Structure of the Code

If you are used to conventional programming languages, the organization of a Tcl program is somewhat unusual. It consists of four sections:

- *Definition of variables* These are variables that are global in scope and of major importance in the program, such as the operating system of the hardware platform (Windows, Linux or Mac).
- *Definition of procedures* These procedures are called from other locations in the program.

³The only difference in this program is the naming of the serial ports.

⁴More precisely, it appears to the user to be interpreted. In fact, there is a run-time compiler that optimizes code execution for speed.

- *Definition of the Graphical User Interface* This code establishes the hierarchy, shape and appearance of the user interface, with labels and variables specified into, for example, a rectangular grid. The `pack` command then invokes the GUI.
- *Start of the Program* This code initializes the system by opening a USB serial port to the hardware and sets up the event handler (the `fileevent` command) for messages arriving from the serial port. Every time a message arrives, the event handler passes its received string to the parsing procedure. The parser then updates various variables, which then automatically update on the display.

This section of the program is usually only a few lines of code. The program is *event-driven*, so clicking on some widget or receiving data from the serial port will be handled automatically.

Reading the Source Code

The source code for this program is in the file `sqm-display.tcl`. It can be viewed and modified in any editor, but is best viewed in an editor that is *tcl-aware*. Then the keywords of the Tcl language are highlighted.

To modify the code, save a backup copy first. Then edit the code and rerun it. Compiling is not required.

Appendix: Terminal Communication with the SQM-LU

Terminal Operation Under Windows XP using Hyperterminal

- Plug in the hardware.
- Determine the assigned COM port as described above.
- Start Hyperterminal.
- To edit any settings, you must `Disconnect`.
- Set 'Connect using' to the correct COM port.
- Set 'Bits per second' to 115200, 'data bits' to 8, 'parity' to none, stop bits to 1, flow control to none or hardware.
- Select `Call` to reconnect.
- Typing in 'rx' should generate a response from the hardware.

Terminal Operation Under Suse Linux using GTKTerm

- Plug in the hardware.
- The command `dmesg` shows the current port assignment.
- Run `GTKTerm`. (Minicom probably will work, but you get to figure it out. I can't.)
- Set port configuration to 115200 baud, 8N1, hardware flow control off.
- Typing in 'rx' should generate a response from the hardware.

Just finished work on a sky quality meter with built in wifi. The device is based on the ambient light sensor TSL2591 and the wifi board ESP32. Communication between the two boards is through I2C. But it should only require one parameter to be adjusted. The code is available on GitHub. Sky-Quality-Meter Here are some pictures. The components: The parts connected: The finished device: This is how output is presented: Sky Quality Data and Analysis with SQM-L. By Captain Magenta. Last year I was given a Unihedron SQM-L, the narrow field of view version of their gadget for measuring night-sky brightness.