DMM Data Logging - NRARAO

May 8, 2014

Steve Olney
North Richmond Amateur Radio Astronomy Observatory (NRARAO)
NSW Australia

Abstract – ***A common requirement for amateur radio astronomy observations is recording the slowly changing output of a receiver. For example, the voltage output of a total power radio telescope in spin scan mode. While there are data logging software packages available, these usually require some specialised hardware interface. This article examines a simpler approach where a common DMM with a data logging capability is used to provide this function.***

I. INTRODUCTION

The re-purposing of consumer products can mean adequate performance at a reduced cost compared with specialised equipment. The DMM is ubiquitous with any electronic store offering a wide range of relatively cheap models – some with data logging functions. Naturally there are some limitations compared to specialised equipment, but as long as these limitations are understood good use can be made of these products.

II. EQUIPMENT USED

For the discussion here two models are used – Brymen BM202 and Digitech QM1571. No statement is made about these being the best examples available; rather these two models were to hand being bought for other projects unrelated to radio astronomy.

III. ADVANTAGES OF DMM DATA LOGGING

While not exactly super cheap, at around AUD$100, at least data logging DMMs are not expensive. For that price you have a multi-function logging interface with overload protection (typically Cat IV 600V), DC and AC functions (voltage and fused protected current), resistance, capacitance and frequency measurements. Compared with typical A/D interfaces this is a wide range of functionality. One data logging DMM can be used for many different projects.

IV. DISADVANTAGES OF DMM DATA LOGGING

A significant disadvantage of using DMMs for data logging is that the timing of the sample data taken is not under the control of the data logging software. For example, the BM202 DMM outputs data at the same rate as the update of the LCD display – approximately every 700mS. There is no pathway for triggering control into the BM202 – the data flow is output only. While this is conceptually unappealing, in practice it is of small consequence.

V. ISOLATION IS IMPORTANT!

A big advantage of using DMMs for data logging is that there are two levels of protection of the PC used for data logging against the effects of high voltages. The first is the Cat IV security class; and the second is the isolation afforded by the data interface. In the case of the BM202 the data interface is optically isolated, while in the case of the QM1571 the data interface is isolated via a wireless[1] USB connection. Not only does this afford protection from damaging voltages but also eliminates interaction via common ground connections and allows true differential mode measurements independent of ground up to the rating of the DMM.

Figure 1: Brymen BM202 (left) and Digitech QM1571 (right)

V. DMM HARDWARE INTERFACES

The two DMM models described here (Brymen BM202 and Digitech QM1571) use different hardware interface types. The BM202 uses an optically-connected serial port connection. The QM1571 uses a wireless connection through a USB port. Nevertheless, examination of the logging software supplied with the QM1571 shows the USB port is actually implemented as an RS232 port. Therefore, both can be accessed by logging software by opening the relevant serial port. Ironically on my particular PC (which has no RS232 port connectors), the BM202 is accessed through an RS232->USB adaptor.

VI. DMM SOFTWARE INTERFACES

For the discussion here two models are used – Brymen BM202 and Digitech QM1571. Both models use a bit-mapped protocol, where the state of each segment on the LCD display is mapped to a bit in the data stream.

The serial protocol for each DMM model is 2400, N, 8, 1. A data packet is transmitted asynchronously every time the LCD is updated by the DMM's conversion cycle. There is no triggering function which can be applied externally – the data comes out automatically[2] without external control.

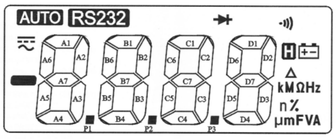
In the case of the BM202 DMM, the data packet consists of 14 bytes of 8-bit data. The mapping is shown in Figure 2. To ensure that the packet data is received as a block, use it made of the invariant 14th byte as an end-sync byte because the MSN (most significant nibble) is always '1110' in binary.

Conveniently, the QM1571 protocol is almost identical. Here

1 Not sure what 'wireless' means – RF or IR link? - would mean opening the 'receiver' case which at this point the author is reluctant to do.

2 The BM202 requires the assertion of the DTR or RTS control lines (or both) to enable the asynchronous data flow.

the length of the data packet is 19 bytes instead of 14 bytes as is the case for the BM202. The QM1571 data packet is basically the BM202 14-byte format is preceded by another 5 bytes – making 19 bytes in total. Monitoring the data stream reveals that, although the 19^{th} byte is also invariant with the MSN = '1110', unfortunately one of the 5 bytes which have been pre-pended occasionally matches the 19^{th} byte (MSN = '1110'). Consequently the first byte (16 hexadecimal) will have to be used in custom logging software instead, as it seems to be invariant and does not seem to occur elsewhere in the data packet.



Output Data set	Bit 7 ~ Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1'st byte	0001	~	⎓	AUTO	RS232
2'nd byte	0010	▬	A5	A6	A1
3'rd byte	0011	A4	A3	A7	A2
4'th byte	0100	P1	B5	B6	B1
5'th byte	0101	B4	B3	B7	B2
6'th byte	0110	P2	C5	C6	C1
7'th byte	0111	C4	C3	C7	C2
8'th byte	1000	P3	D5	D6	D1
9'th byte	1001	D4	D3	D7	D2
10'th byte	1010	u	n	k	⏩ (diode)
11'th byte	1011	m	%	M	·)))
12'th byte	1100	F	Ω	Δ	H
13'th byte	1101	A	V	Hz	(battery)
14'th byte	1110	x	x	x	x

1: Corresponding LCD segment on
0: LCD segment off
X: don't care

Figure 2: BM202 Data Packet Protocol (ref:Brymen "2500-count DMM communication protocol")

The BM202 interface requires that either the DTR or RTS line is asserted to enable the flow of data. Either of these lines can be used to stop and start the flow of data from the BM202. However, the actual timing of the data conversion inside the BM202 still remains under the control of the DMM's internal electronics. The Brymen information states that the data flow is only output when the voltage level of the TXD line is positive and recommends that the TXD line is made negative to inhibit data flow. The custom software did not address this requirement as presumably the idle state of the un-driven TXD line is positive. In any case it is much easier to assert/de-assert the DTR/RTS lines. The QM1571 ignores the DTR/RTS lines and so might be able to be controlled via the state of the TXD line. This suggestion was not tested.

A typical 14-byte data output from the BM202 for an example[3] LCD reading of "AC 218.9V auto-ranging" is...

1BH 25H 3BH 40H 55H 67H 7FH 8BH 9FH A0H BOH COH D4H E8H

The QM1571 19-byte data packet tacks an extra 5-bytes onto this 14-byte BM202 example. Note how the MSN increments from 01H to E0H (this applies for both model DMMs' patterns). That incrementing pattern could be used to assemble the packet data instead of synch-ing to the E8H terminating byte in the case of the BM202, or the 16H header byte for the QM1571.

VII. CUSTOM LOGGING SOFTWARE

Each model DMM comes with its own data logging software from the manufacturer. While the BM202 data logging software allows saving logged data into a text file, the QM1571 software throws an exception when attempting to save a data file. This may be due to the software not being compatible with Windows 7 Pro. The solution adopted by the author is to abandon the supplied manufacturers' data logging applications and write custom data logging software for each DMM which allows saving of the data in any format required.

3 Reference: Brymen "2500-count DMMs communication protocol"

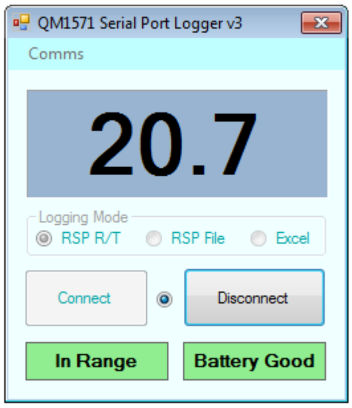


Figure 3: Custom Data Logging Software for QM1571 DMM - reading temperature

The custom data software takes the bit-mapped data packet and de-obfuscates the bit patterns to determine the value displayed on the LCD screen of the DMM. Adjustments are made to the value based on the presence, or otherwise, of the negative sign segment and the position of the decimal point segment. The value is converted into a string ready for display in the custom software and available to be saved in a data file. The data can be time-stamped if needed. The data thus saved can be imported into other programs such as Excel or Radio SkyPipe II Pro.

Included on the custom software GUI are status panels for over-range and battery condition as shown in Figure 3.

There are three 'Logging Mode' settings.

- RSP R/T – Radio SkyPipe II real-time mode. The data is written to a text file (no time stamping – provided by RSP) and then a second dummy file is created ('OK.txt') whose presence is used as a 'gate' to prevent a clash with writing/reading of the data file. After the data file is read, both files are deleted. The next appearance of a 'OK.txt' dummy file starts the reading cycle of the next data value..
- RSP File – Radio SkyPipe II offline mode. The data is written to a text file with time-stamping in a format suitable for importing into RSP at a later date.
- Excel File – Spreadsheet offline mode. The data is written in CSV file format with time-stamping suitable for importing into a spreadsheet at a later date.

Note: For the first logging option (RSP R/T) the bridge between RSP and the custom data logging program is an UDS (User Defined Source) connection. In the case used here by the author the bridge application is FileUDS[4]. It is not the author's intention to detail the procedure for achieving this and it is left to the reader to investigate.

VIII. SAMPLE DATA RUNS

An example of each of the three logging modes is given below for the QM1571 custom data logging software. Note that the 2^{nd} and 3^{rd} options (RSP File and Excel File) are off-line modes and so require time-stamping by the custom data logging software. The first mode (RSP R/T) is real-time so the time stamping can be conveniently left to RSP where either a local time stamp can be applied or, especially relevant to radio astronomy, a Local Sidereal Time (LST) or UTC time-stamp

4 It is advisable to get the latest version of FileUDS from the RSP site. Googling "Radio SkyPipe" will return the latest references to RSP. Look for the UDS section for the zipped compilation of sample UDS executables and example code. FileUDS is contained within that zip file.

can be applied.

Each of the three examples log data from the QM1571 DMM reading a K-type thermocouple to which a finger is applied and withdrawn to simulate a slow rise and fall of some parameter. The custom software was temporarily modified so as to output all of the three modes data at the same time so that the same data is shown in each of the example runs.

The sample rate (LCD update rate) is about two times per second as shown by the timestamps on the sample data file text given.

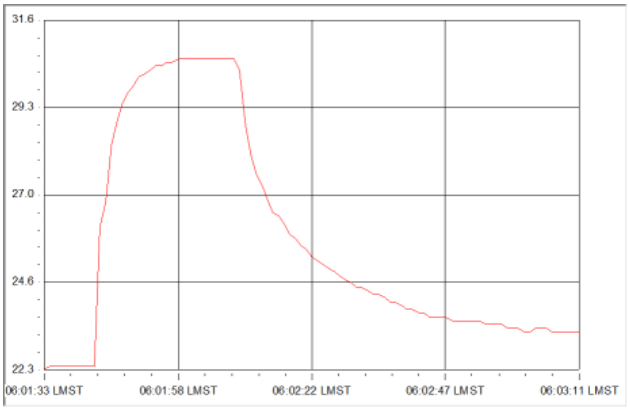


Figure 4: Output chart from RSP II when custom software is operating in 'RSP R/T' mode.

The chart produced by RSP when operated with the custom software in 'RSP R/T' mode (real-time) is shown in Figure 4. Note that RSP is in LST time stamping mode.

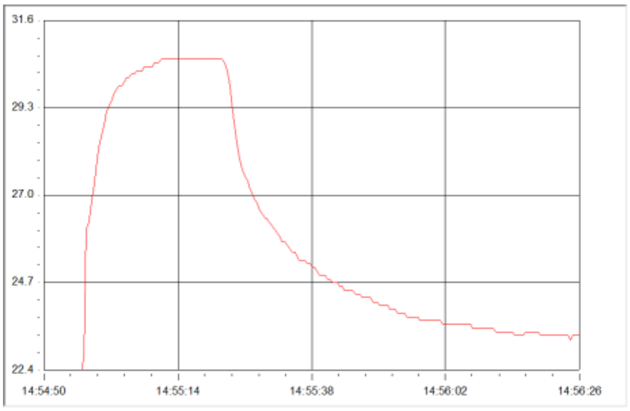


Figure 5: Chart produced by RSP II after importing the data text file.

The chart produced by importing the data into RSP II via the text file import process is shown in Figure 5.

The format of the data file is shown here...

08/05/2014	14:54:50.219	022.4
08/05/2014	14:54:50.669	022.4
08/05/2014	14:54:51.119	022.4
08/05/2014	14:54:51.569	022.4
08/05/2014	14:54:52.018	022.4
08/05/2014	14:54:52.468	022.4

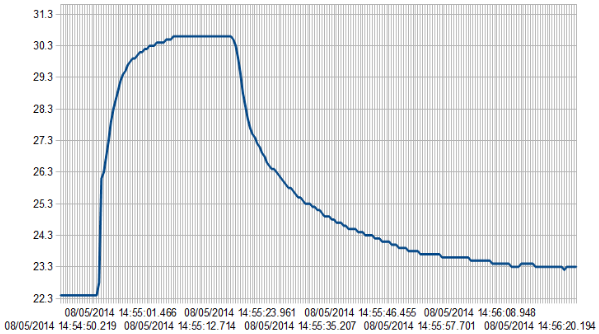


Figure 6: Chart produced by importing CSV data file into Open Office Calc.

Importing the CSV data file into Open Office and charting the data produces the graph as shown in Figure 6. The format of the data file is shown here...

08/05/2014 14:54:50.219,022.4
08/05/2014 14:54:50.669,022.4
08/05/2014 14:54:51.119,022.4
08/05/2014 14:54:51.569,022.4
08/05/2014 14:54:52.018,022.4
08/05/2014 14:54:52.468,022.4

IX. CONCLUSION

For slowly varying voltages it is possible to use consumer-grade DMMs for data logging purposes.

While not cheap, for the price it is possible implement a multi-purpose logging interface with good isolation for protection of the logging PC and eliminate a source of ground loop noise.

The writing of custom software allows a flexibility of how the data is saved. The data can be formatted to suit a range of data display applications.

The ability to record data in real-time through the custom software interface then passing through the File UDS bridge into RSP II means that data can be time stamped with local, UTC or LST time.

The setup described here is suitable for such radio astronomy projects as drift-scan observations. For a 12GHz radiometer in drift-scan mode equipped with a 2m dish, the time for the approximately 1 degree HPBW to pass over a point source object is about 4 minutes. During that time more than 250 data points would be taken – enough to plot the curve with sufficient accuracy.