• E-Mail: mike@pwpublishing.ltd.uk

Getting Started in Data Modes

Fresh back from the RSGB Convention, Mike Richards G4WNC feels that it's time to go back to basics for amateurs starting to get interested in the possibilities of what the many different data modes have to offer.

I've just returned from the very successful RSGB Convention at Milton Keynes. As usual, I was roped-in as the official photographer and also provided lectures on the Raspberry Pi and Data Modes. I delivered both lectures on the Saturday and Sunday and had good attendance at each session. Both of my presentations along with supporting information are available via the '2015 RSGB Convention' tab on my website, below.

g4wnc.com

From talking to delegates at the convention, there are clearly lots of people who have considered data modes but need some help to get started. I therefore thought it might be timely to include some getting started information in this month's column.

Data Modes – Getting Started

One of the great things about modern data modes is that just about every mode is computer generated. The obvious benefit of this is the ease with which you can change modes. In many cases, the data modes software itself will support many different data modes but for a different set of modes, you simply change to a different software package. Data modes software is free and available for most computer operating systems, including Android and IOS, but Windows users enjoy by far the largest selection.

Data Modes Transmission

With all data modes being generated in the computer, the link between the computer and your rig becomes very important so let's see what's required. Although many modern data modes use quite sophisticated modulation systems, they all emerge from the computer as audio tones. So how does that work? The secret is in the use of a standard SSB rig set to USB mode. In the absence of an audio input, an SSB rig should emit no signals at all other than perhaps a very small amount of carrier leakage. If we inject a 1kHz audio tone into the microphone socket of an SSB rig tuned to 14.1MHz, the output should be a single RF frequency that is the carrier frequency itself plus 1kHz, in other words (14.100MHz + 0.001MHz) = 14.101MHz, **Fig. 1**. If we change to a 2kHz audio tone, the output should be 14.102MHz. Now, many of the data modes systems use variations of frequency shift keying (FSK) where the carrier is shifted to convey the data information. By using an SSB rig, all we have to do is to produce a modulated audio signal and the rig will translate it into an RF frequency shift. Similarly, when using phase shift keying (PSK) systems, we just have to phase modulate an audio signal and send it to the microphone input of the SSB rig. This form of PSK and FSK emulation is called audio frequency shift keying (AFSK) or audio phase shift keying (APSK).

Many data modes require the SSB rig to operate in linear mode so it's important to adjust and monitor your power amplifier (PA) drive levels very carefully. A simple technique is to put your rig and data modes software into transmit with a dummy load connected and adjust the drive until the automatic linearity control (ALC) just starts to kick-in, then back-off slightly. You will also need to keep an eye on this adjustment when transmitting. While talking about the transmit side, you should note that you will only need very low powers when operating data modes. Most rigs are not designed to supply their full output when used with 100% duty cycle modes so it is common to limit the power on data modes to a maximum of 30W (in the case of a typical 100W transceiver). However, in practice, you will need far less than this and I normally operate with around 5W into my Butternut HF9V vertical antenna. An added bonus of the lower power is a much lower chance of TVI or other interference problems.

Data Modes Interface

Because all data modes operation will be using the audio link between the computer and the rig, it's worth spending some time to get it right. The simple solution is to go and buy a data interface from your favourite dealer. However, if you're prepared to do some homebrew, it's very easy to make your own interface and I've provided links to some designs at the end of this section. I've also shown a block diagram of a basic system in **Fig. 2**. Many people prefer to include some isolation between the rig and the, potentially very noisy, PC. The simplest way to do this for the audio lines is to use transformer coupling. Line transformers are ideal and these are readily available from suppliers such as Farnell and RS Components.

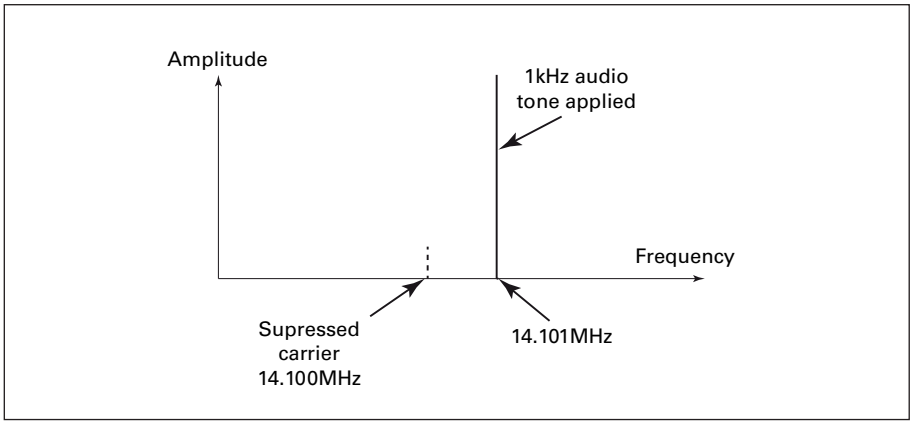

Fig. 1: Illustration of USB output with a 1kHz audio tone applied.

Macro Key Name	Macro	Description
CQ	<TX> CQ CQ de <MYCALL> <MYCALL> <MYCALL> pse k <RX>	Switches to Tx and sends a short CQ call and returns to Rx
CQ Ans	<TX><CALL> de <MYCALL> <MYCALL> <MYCALL> kn <RX>	Switches to Tx sends his/her call followed by your call three times and switches to Rx
In QSO	<TX> <CALL> de <MYCALL>	Switches to Tx and sends his/her call followed by your call and stays in Tx
RST	rst <RST> <RST> name: <MYNAME> <NAME> qth: <MYQTH> <MYQTH> loc: <MYLOC> <MYLOC>	Expects the rig to be in Tx mode and sends the RST plus, name, QTH and locator
Station	Plain text	This will only be used occasionally and should include brief details of your rig, antenna and data modes software
KN	btu <NAME> <CALL> de <MYCALL> k <RX>	This used at the end of your over and will return the rig to Rx after sending the call signs
SK	Tnx fer qso, good dx <NAME>, 73 <CALL> de <MYCALL> sk <RX>	This is the end of QSO message.

Table 1: Suggested Starting Macros (FLDIGI)

You'll see that I've included a level control in the transmit audio path of Fig. 2. I have found this to be essential because you often need to tweak your drive levels for each band and having a control knob to hand is much simpler than using software level controls.

When it comes to Tx/Rx switching, there are a few options. The simplest is to use the voice operated transmit switching (VOX) that's available in most rigs. All the popular data modes software keeps the transmit audio output silent until you switch to transmit. This fits very well with VOX operation and means the PC-to-rig connection can be simplified to a pair of audio leads with a level potentiometer in the transmit side, again as shown in Fig. 2. While VOX is fine for a basic data modes station, many data modes packages have the ability to interact with the rig's standard CAT (computer interface) controls to manage Tx/Rx switching along with storing and controlling the operating frequency. As a result, many operators prefer to use a CAT link to the computer. This is another area that can be self-built and the best way to achieve isolation in a CAT lead is to use an opto-isolator. These are cheap and readily available from most good component sellers. If you've decided to buy a ready-made commercial interface unit, you can use the information learnt here to check that it provides the facilities you want.

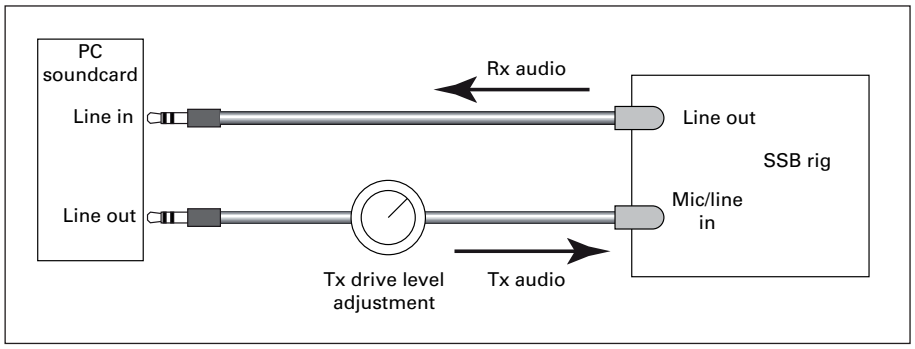

Fig. 2: Basic data modes interface.

If you want to be bang up-to-date, you could use a Bluetooth interface to provide the audio connection between the rig

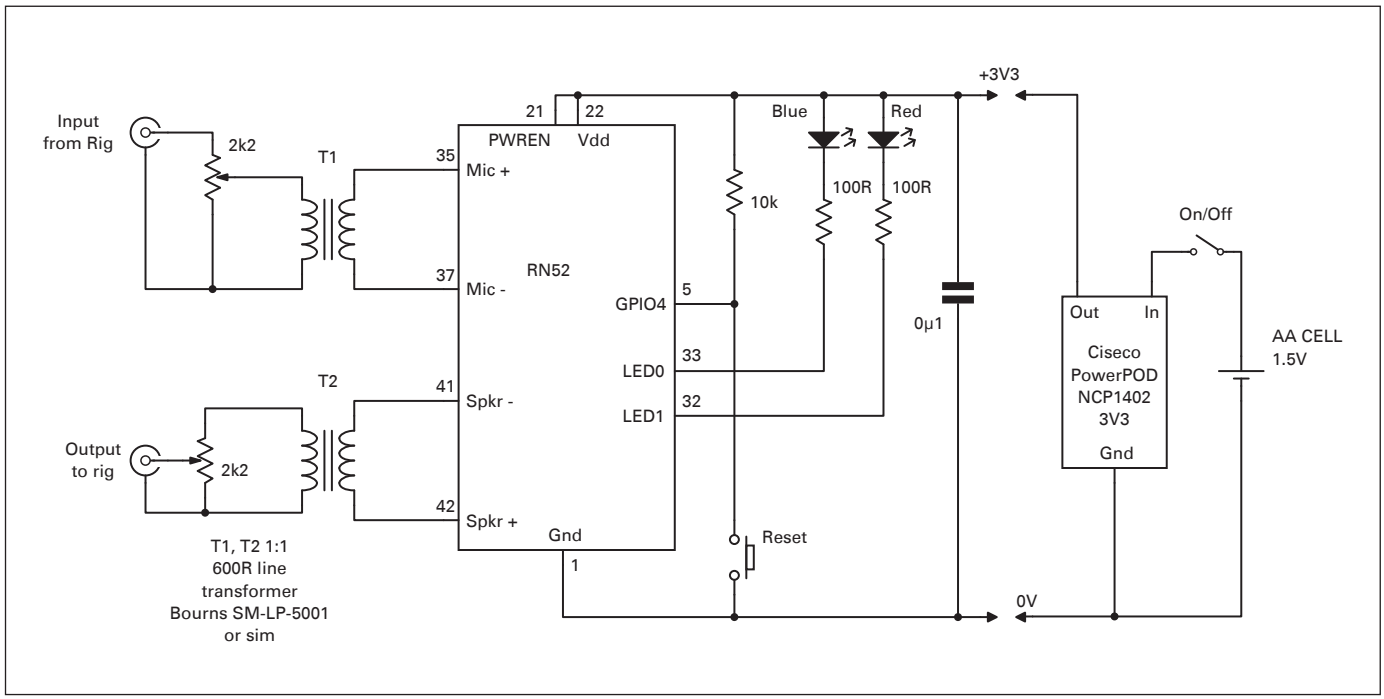

Fig. 3: Diagram of a Bluetooth interface based on the MicroChip RN52 module.

and the computer. You can even make your own Bluetooth interface using the MicroChip RN52 Bluetooth audio module. These are around £15 each and only need a few simple connections. I've shown a circuit diagram in **Fig. 3** and I covered this unit in more detail back in the June 2014 issue of *PW*. If you enjoy hacking, you could repurpose a Bluetooth headset as a data modes interface.

www.wa1wa.net/filespdf/pskhandbook.pdf

www.tdars.org.uk/html/2009-10/Data%20Modes/PSK31%20circuit.pdf

Typing Nightmare

Once you have your interface set up, the world of data modes is literally at your fingertips. Now, one of the points I picked up at the weekend is that most of us are not that quick on the keyboard and the added pressure of trying to type a live QSO can be a step too far for some. Don't worry, you're not alone and help is at hand! Many data operators suffer the same problem but there is a simple software solution in the form of macros. In data modes terminology, a macro is a short item of stored text that can be recalled with a single button press. In order to relieve the operator of the typing burden, it's common practice to use macros for all the main parts of a QSO. In **Table 1**, I've shown some suggested macros that you can use to get started. However, you will find that many of the data modes software packages have example macros already installed so you just have to modify them to suit your operating style. In addition to being great time savers, modern macros have a few tricks designed to make life even easier. Using FLDIGI as an example, **Fig. 4**, you will notice the use of arrow brackets around some parts of the macro such as <TX>, <MYCALL> and <CALL>. These are special fields in the macro that are automatically populated from the data you entered when you set up the program but they can also use information gathered from the station you are working. In FLDIGI, the <MYCALL> field is populated with the callsign you entered for your own station. Two other special fields in the macro are <TX> and <RX>. These can be used to automatically switch to transmit and receive.

In addition to the availability of macros, most modern data modes software can identify key elements from the station being received and transfer them to the macro. Let's look at FLDIGI receiving PSK-31 as a common example. In **Fig. 5**, I've shown many of the important areas

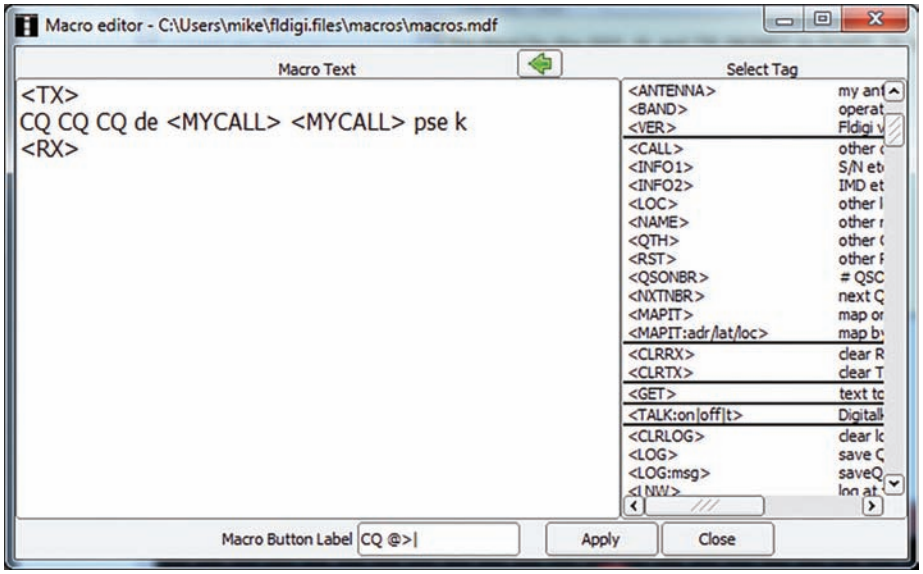

Fig. 4: FLDIGI macro showing the special characters.

of the FLDIGI user interface. When you decode a *CQ* call from a station you want to work, you first move your cursor over the received callsign and left-click. You will see that the callsign is transferred to the Call box in the logging section of the FLDIGI at the top of the screen. To answer the *CQ* call, use the ANS macro and the rig will automatically switch to transmit, send his call followed by your call three times and switch back to receive – all with one button press. Once the QSO is in progress, you can use the same technique to capture and store vital information such as operator name, grid square, QTH and signal report. Using this simple technique, it's possible to hold a complete 'rubber-stamp' QSO with almost no typing. Once you've got started in data modes, you can gradually customise the macros to fit your operating style. One final point – when you start building your own macros, don't forget to save them or they will be automatically reset when you restart the software.

Summary

As I hope you can see, getting started in data modes is easy enough and with a small amount of homebrew, can be done at very low cost. When compared with SSB operation, data modes have many benefits such as greatly reduced bandwidth, lower power requirement, macros that overcome language barriers and many more.

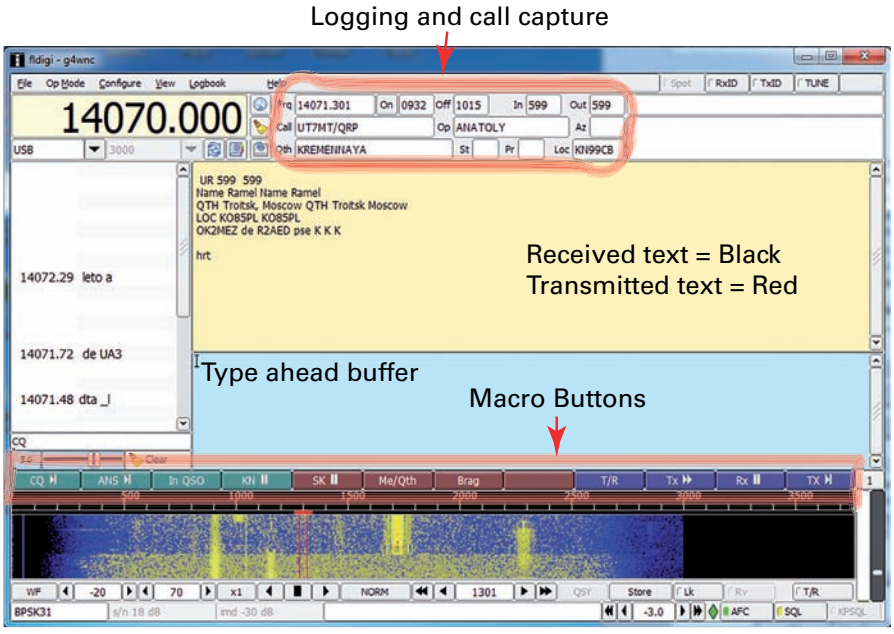

Fig. 5: FLDIGI main screen layout.