Mike Richards, G4WNC • e-mail: mike@photobyte.org

The Raspberry Pi in your shack

An ultra-cheap computer with a wide range of amateur applications

PRACTICAL PI. At last year's RSGB Convention I gave a presentation on practical uses of the Raspberry Pi computer in the shack. The talk was well received and in this article I will cover some of the projects and bring you up-to-date with the Raspberry Pi.

Although primarily designed as an educational tool to encourage youngsters into computer science, the Raspberry Pi has aroused a huge interest from the Maker fraternity. Its compact size, power and attractive price makes it an ideal powerhouse for a lot of small scale computing projects. The original designs of the model A and B were only expected to sell a few thousand but sales are currently well over 4 million!

RASPBERRY PI RANGE. There are currently five models in the Raspberry Pi line up, namely the Model A, Model A+, Model B, Model B+ and the Compute Module. The Compute Module is a very different beast that has all the computing power of the Pi but arranged in a computer memory strip format with lots of additional connections. It is specifically designed for OEMs so I will not cover it here.

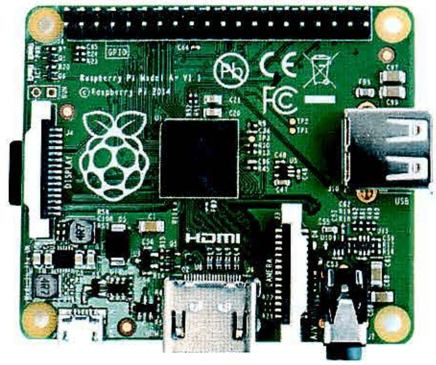

PHOTO 1: The Raspberry Pi Model A+ with new PCB layout.

The new Plus models were introduced in the latter part of 2014 as compatible improvements to the original model A and B. The new and improved models have many new features that make them more attractive and are the first choice for many new projects. The Model A is the simplest and cheapest Raspberry Pi and it comes without an Ethernet port and with a single USB port. As a result, it is lighter, cheaper and has lower power consumption than its bigger brother. The original Model A was the same physical size as the Model B but had a smaller memory and some components omitted. However, the new Model A Plus (Photo 1) uses a new PCB layout that, at just 57mm x 68mm, is much smaller and lighter than the original. It also features an extended, 40-pin general purpose input/output (GPIO) connection, micro-SD card slot and a more efficient power supply. The Model B+ is also significantly improved with 4 USB ports (originally 2), 40-pin GPIO (originally 26), improved power supply, micro-SD operating system card and evenly spaced mounting holes. Unless you have a low power project that doesn't need Ethernet or multiple USB ports, the Model B+ is the model with the most appeal to hobbyists and experimenters. Let's now take a look at some practical uses around the shack.

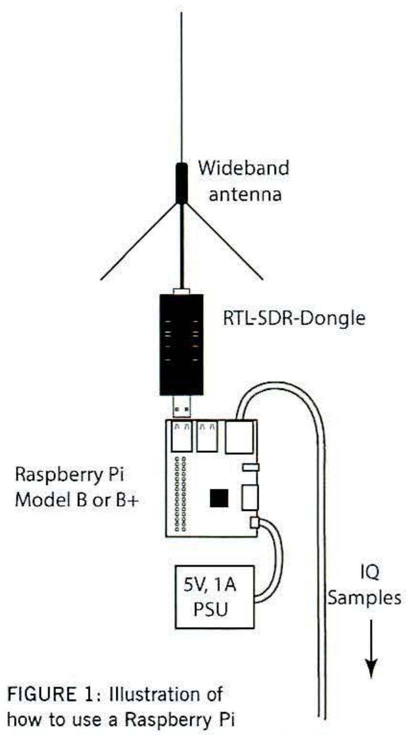

FIGURE 1: Illustration of how to use a Raspberry Pi to stream IQ samples from a dongle.

SDR RECEIVER SERVER. In this project I will show you how to stream a remote mounted SDR dongle back to the shack. The DVB-TV dongles based on the RTL2832u chip provide a very cost effective way to create a wideband SDR receiver. These generally have a frequency range that extends from around 60MHz up to well over 1GHz. At the higher frequencies, feeder loss can become a real problem so there can be a significant advantage if the receiver is located close to the antenna. I've shown a diagram of the proposed system in Figure 1. This is not a new idea but it is a technique that has been broken since autumn last year. The problem has arisen since the introduction of the B+ model and appears to be linked with libusb in the Raspian Linux distribution and is yet to be resolved. However, I've been experimenting with the Pidora version of Linux and will show you how to set up a working SDR dongle server using that operating system.

Pidora Setup. The easiest way to set up your Pi is to use an Ethernet network connection with an HDMI monitor and a keyboard/mouse combination. I use the Perixx Periduo 707 Plus Wi-Fi keyboard and mouse with all my Pi projects as it's cheap, works well and is compatible with all Pi models. Before you start you will need an SD or micro SD card loaded with the Pidora operating system. You can download a Pidora image from the download section of the Raspberry Pi home site. The image is supplied as a zip file so you will need to extract it to a directory of your choice on your PC. If you're using Windows, you will also need to download the Win32 Disk Imager so that you can burn the image to your SD card. You will find full instructions for burning the operating system to an SD or micro SD card at [1].

With the SD card prepared, you are ready to start the configuration. Insert the card into the Pi with the contacts facing towards the PCB. You can then apply power and you should see signs of life, with LEDs flickering and text on the screen. On its initial start-up, Pidora runs a wizard that will take you through the essential configuration steps. This begins with licence agreements followed by the keyboard language and then the creation of a user ID. As we're not too worried about security for this application, I suggest you stick to using 'pi' as the username and 'raspberry' as the password for this screen and also the root password on the next screen. NB: when you create the user make sure you tick the 'Add to Admin group' box. Failure to do this will mean you can't use the 'sudo' command later. You can ignore the Network login and Advanced buttons. When you get to the time-zone

Mike Richards, G4WNC • e-mail: mike@photobyte.org

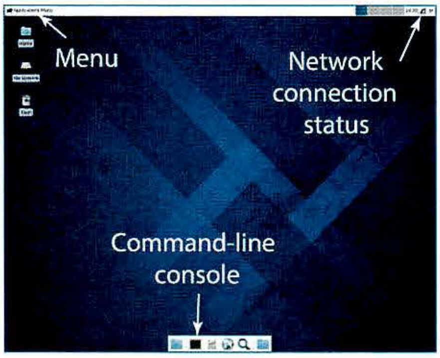

FIGURE 2: Pidora start screen.

screen, select the UK by choosing Europe – London. Leave the file system settings at their default. On the System Settings screen make sure you tick the HDMI to DVI adapter box if you're using an adapter cable. If you don't do this and are using a DVI adapter there will be no screen output after the system reboots – I know because it happened to me!

That completes the setup of the Pidora operating system and, following reboot, you should end up with a screen similar to that shown in **Figure 2**. When prompted, accept the default console settings.

The next step is to begin loading the software and libraries required to support the SDR dongle server. If you've previously used the Raspian strain of Linux you will encounter a few different commands with Pidora. The first step is to open a command line console by clicking the black screen icon at the bottom of the display. The first point to note is the tool we use to install packages which is called 'yum' in place of 'apt-get' that's used with Raspian. To install packages you will see that I've used yum with the '–y' option. The '-y' option effectively bypasses all the y/n prompts that occur during a normal installation and saves unnecessary keyboard interaction.

During yum installations, you should see a progress report on the screen listing all the actions which will include automatically downloading and installing a range of support software.

Now that everything is ready, let's start installing software packages by entering the commands in **Table 1**.

The prerequisites we need are 'cmake', 'libusb-devel' and 'git'. These are downloaded using the first three lines of the text in Table 1. The third line, git, is required so we can clone the rtl-sdr software from the Osmocom online repository. If you don't want to sit by the terminal watching the download and entering commands, it's very easy to create a simple script file that will run the commands for you. Here's how to do that:

1. In the command-line console type: 'cd ~' to make sure you are in the home directory.

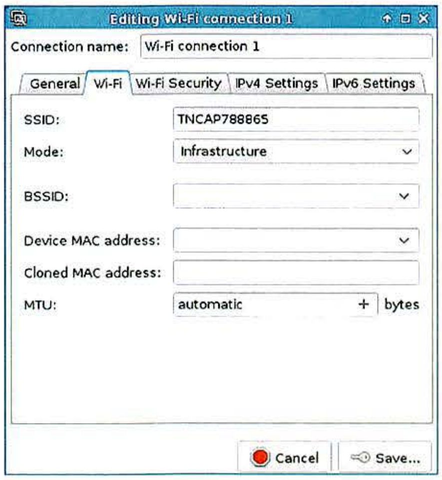

FIGURE 3: Pidora Wi-Fi configuration panel.

2. Type: 'nano sdr.sh'. This will create a new file called sdr.sh and open an edit screen.
3. Type in the commands exactly as shown in **Table 1**.
4. Type control x and answer y to the prompt followed by return. This will save the file.
5. In the command-line console type: 'sudo chmod +x sdr.sh'. This will change the permissions of the file and make it executable.
6. To run the script type: './sdr.sh'. You will be asked to enter your password (raspberry) at the beginning and once more just before the end.
7. This will take a while so make a cuppa!

The final task is to create a configuration file that will tell the system where to find the libraries we've just installed. To do this we first create a new file:

'sudo nano /etc/ld.so.conf.d/sdr.conf'

That will create a new file called sdf.conf and open it in the nano editor so we can add the following, single line of text:

'/usr/local/lib'

Save the file by pressing ctl X followed by 'y' then return. The final task is to update the system path by typing:

'sudo ldconfig'

That completes the installation so you can now reboot by typing 'sudo reboot' in the console.

Once the system has rebooted and you have entered your password, you can check the operation of the SDR dongle server. To do this, open up the command-line console again and type 'rtl_test' This will report the dongle device, list the tuner type and start a continuous read from the device at 2.048MSPS. To complete the test press control C and you will see a report showing the number of samples lost, which is usually 0. That completes the configuration of the Pi, so the next stage is to prepare it for remote operation as a server.

Pidora Wi-Fi Connection. Whilst the SDR streaming works at its best with an Ethernet connection, for some, a Wi-Fi connection may be the preferred route. The disadvantage of Wi-Fi is a much slower network connection, which means the sample rate of the dongle needs to be reduced, which, in turn, gives a reduced IQ tuning band. Here's a step-by-step guide to adding a Wi-Fi connection to Pidora:

1. Open the Application menu (top left) and select Settings – Network Connections and click the +Add button.
2. Choose Wi-Fi from the drop-down menu and click Create, see **Figure 3**.
3. On the Wi-Fi tab enter the SSID of your Wi-Fi network – you should find this printed on your router case.
4. Select the Wi-Fi Security tab and choose the security system (usually WPA & WPA2 Personal) then enter the Wi-Fi password for your system.
5. Click Save and, if prompted, cancel the request for a Keyring password.

That completes the Wi-Fi setup and the Pi will remember the details and automatically connect when your Wi-Fi is within range.

Headless! For use as an SDR Dongle server, we want the Pi to operate as a standalone, network connected unit with no screen, keyboard or mouse. This mode of operation is known as 'headless' and we control the Pi from a remote PC via a protocol called Secure Shell Tunnelling (ssh). It sounds complicated but is dead easy to use.

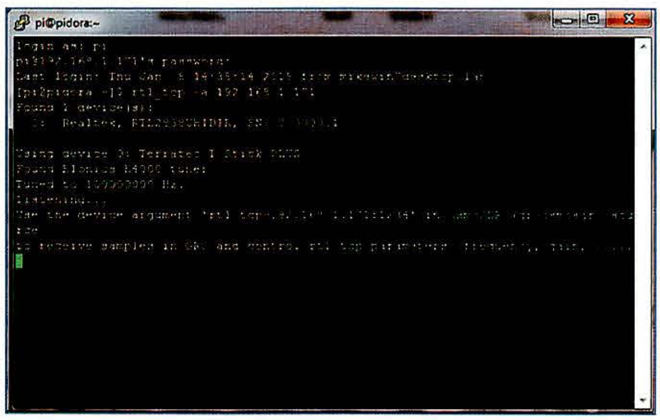

FIGURE 4: Remote access terminal showing the Pi ready to accept an *SDR Sharp* connection.

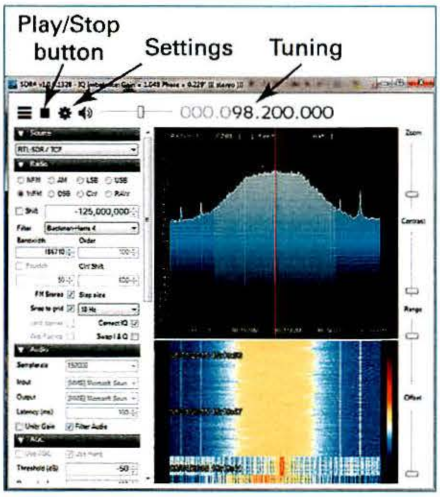

FIGURE 5: *SDR Sharp* **main screen with key controls identified.**

All we need on the main computer is a simple terminal application such as the old Windows *Hyperterminal* or one of the other open source solutions such as *PuTTY* or *Termite*.

To start a new connection we need the IP address of the Pi. As you probably know, the IP address in a standard home network is allocated by your router using its DHCP server. Ideally we want the Pi to be allocated the same IP address every time it starts and the method of achieving this will depend on your router. One of the simplest solutions, which is available on most home routers, is to force the router to allocate the same IP address to specific hardware. With this option set, the router will see the hardware details (MAC address) of the Pi's Wi-Fi dongle or Ethernet chip and automatically allocate the reserved IP address. The alternative solution is to use a static IP address for the Pi. To do this you will need to consult the manual for your router and follow the instructions for allocating a static IP address.

If you want to see the IP address of the Pi whilst you still have the keyboard and screen connected type: 'IP addr show' at the command line. The first entry shown will be the Pi's loopback address which will show as 127.0.0.1 but you need the eth0 entry for a wired connection or wlan0 for a Wi-Fi connection. The IP address will consist of four numbers separated by dots, usually 192.168.n.n, where n is specific to your own local network.

Once you've established the connection method and the IP address, you can create a headless file for the Pi. This is a simple text file that sits in the root directory of the Pi's SD card but triggers a couple of useful actions. Here are the steps to create a headless file:

1. Power down the Pi.
2. Plug the Pi's SD card into your main PC.
3. Open a simple text editor such as Notepad.
4. Enter text as shown in **Table 2** for a DHCP IP address or Table 3 for a static IP address.
5. Save the file to the root of the SD card with the name 'headless' and no suffix.

In addition to carrying out some useful configuration, the presence of the headless file causes the Pi to read out its IP address via the headphone socket once it's finished booting. The IP address is also sent by flashing the activity led. If you have a smartphone, another way to find IP addresses on a network is to use the free app *Fing*. This will report all the devices on your network and can be very useful.

Get Streaming. That completes the Pi setup so you can start using the system. I suggest you start with *SDR Sharp* as it's currently the simplest system to use and includes full support for RTL based dongles. You can download the latest version from **[2]**.

Here are the steps required to start receiving IQ samples from the remote Pi and SDR dongle:

1. Power up the Pi.
2. Open your main PC's terminal software, eg *Hyperterm, PuTTY* or *Termite*.
3. Choose an SSH connection, enter the Pi's IP address and click Open or Connect.
4. You may see a security warning; accept or cancel that.
5. You will see the log in prompt for the Pi. Enter your username password, which will be 'pi' and 'raspberry' if you followed my earlier instructions.
6. With the login complete, type rtl_tcp –a 192.168.1.171 to start the server. NB: replace the IP address with the one allocated to your Pi.
7. You should see a screen similar to **Figure 4** showing that it has found the dongle and is waiting for a connection.
8. Open *SDR Sharp* on your PC and select RTL-SDR/TCP as the Source in the top left of the screen (**Figure 5**).
9. Click the cogs icon and set the host to the IP address of your Pi and leave the sample rate at 2.048MSPS.
10. Click the Start button (arrow) at the top left of the *SDR Sharp* screen to start the receiver.

The display should burst into life and there should also be some audio signal. If the audio is choppy, that's an indication that your network can't handle the sample rate. For a Wi-Fi connected network you may find that you have to reduce the sample rate down to 0.25MSPS. You can alter the sample rate by opening the cogs icon and using the drop-down menu to choose a slower rate. Ethernet connections are usually fine at 2.048MSPS. For best results you will also need to adjust the RF gain and AGC settings. A good start point is to tick the Tuner AGC once the receiver is running. For more help with operation of *SDR Sharp* you will find lots of tutorials on the web.

LOCATING THE PI. The best location for your Pi will depend very much on your QTH but the loft often makes a good starting point. The precise power requirements will depend on the type of network connection and which model of Pi you are using. My measurements with an Ethernet connection and a Terratec dongle show a current draw of 450mA with the Model B+ and 600mA with the older Model B. If you do decide to mount the Pi outside, you will need a good weatherproof box with ventilation to prevent overheating in warmer weather.

NEXT TIME. Next time I'll be showing you how to use the Pi for datamodes and press it into service as a security camera or wildlife monitor. If you would like to use your Raspberry Pi with a pre-loaded and tested SD card, I have these available for sale via my website **[3]**. Each card is individually tested and supplied with full instructions.

WEBSEARCH

[1] Loading OS to memory card – http://goo.gl/hKmGhe
[2] SDR# download – http://sdrsharp.com/#download
[3] My website – www.g4wnc.com

TABLE 1: Pidora commands to install the rtl-sdr server software.

```
sudo yum -y install cmake
sudo yum -y install libusb-devel
sudo yum -y install git
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr
mkdir build
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON
make
sudo make install
```

TABLE 2: Content of the 'headless' file without a static IP address.

```
RESIZE
SWAP=512
```

TABLE 3: Content of the 'headless' file with a static IP address. NB: change the IP address and Gateway to match your system.

```
IPADDR=192.168.1.171
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
RESIZE
SWAP=512
```

Mike Richards, G4WNC • e-mail: mike@photobyte.org

The Raspberry Pi in your shack

Using the brand-new Raspberry Pi 2 Model B

Originally Presented at RSGB 2014 Convention *See You in 2015*

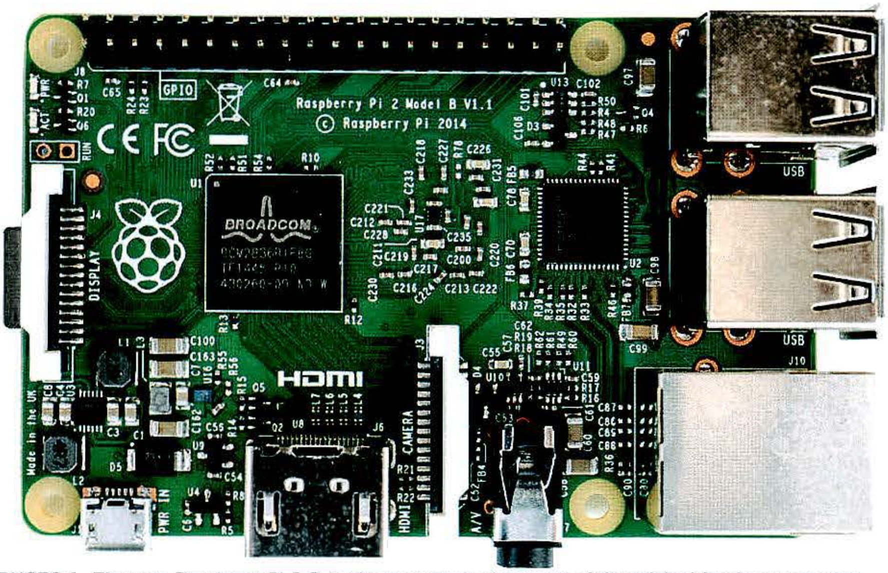

PHOTO 1: The new Raspberry Pi 2 B is about six times the speed of the original for the same price.

RASPBERRY PI 2. Hot news for this month is the surprise release of the new Raspberry Pi 2 Model B (**Photo 1**). I knew that something was in the pipeline and had assumed it would be the long awaited touch-screen display unit – I was clearly wrong! The new Pi 2 sells at the same price as the previous models but provides a significant performance upgrade that makes it a lot more attractive for amateur radio use. The main change has been the use of a new 900MHz quad-core ARM Cortex-A7 central processor. This is a significant power boost over the old, single core, 700MHz ARM unit. The new processor is supplemented by an increase in the RAM, now 1GB as opposed to the 512MB in the B and B+ (and 256MB in the early Model As). The net result is a new Pi 2 that is a good six times faster than the B+. The Pi 2 is now starting to look like a respectable little computer in its own right so it should be able to take on many of the tasks that we couldn't get near with the previous models. I ordered my Pi 2 as soon as it was announced and can confirm that the new processor has totally transformed the general feel of the Pi. The web browser is now slick and all the software I've tried has been transformed. Boot time to the GUI is also much quicker, at just over 20 seconds. The other significant development is a link-up with Microsoft that promises a free copy of Windows 10 to Pi 2 users that sign up as a Windows IoT developer.

DATAMODES WITH THE PI. Using the Pi for datamodes is actually very straightforward and here I'll take you through the steps required to install and configure the software. The Pi can provide all the computing you need to run a portable/QRP station and the Pi 2 could even be pressed into service as the main shack PC!

The first step is to make sure you have the necessary hardware. The first requirement is some sort of keyboard and mouse combination. There are lots to choose from and you may well have a spare USB mouse/keyboard combination kicking around. I've been using a Perixx compact, wireless, keyboard and mouse combination that has served me very well for the last 3 years. The model I use is the Periduo-707 Plus that sells on Amazon for around £19. This is fully compatible with the Pi and uses a couple of AA cells for the keyboard and two AAA cells in the mouse. The batteries seem to last a long time and I've dropped my keyboard many times and it still keeps working!

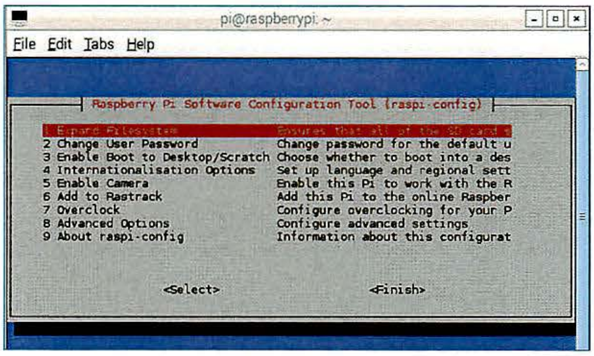

FIGURE 1: The Raspberry Pi configuration utility.

Next on the list is a USB soundcard. Although the Pi has HDMI audio and a 3.5mm audio output jack, it doesn't have an audio input jack. Fortunately, this is very easily solved as there are lots of USB soundcards on the market. If you're buying new, I suggest you check the Raspberry Pi compatibility list that you'll find at http://elinux.org/RPi_VerifiedPeripherals . I have been using the Daffodil US01 soundcard (**Photo 2**) that's available from Amazon and other suppliers for around £5. You will also need a short USB extension cable for the sound card because its bulky body often obstructs the other USB ports.

You will also need a display of some kind and I strongly recommend using an HDMI monitor. This is because the video quality from the HDMI output is far superior to the Pi's composite video output. There is a very wide choice of monitors on the market ranging from standard PC monitors to small touch screen units. You can also make use of a standard TV, providing it has an HDMI input.

Finally, a power supply is required. I measured the current consumption of models A, A+, B, B+ and the new Pi 2 in **Table 1**. This shows the maximum power draw whilst decoding PSK-31 with a Daffodil soundcard and an HDMI computer monitor. As you can see from the results, a 1A 5V power supply should be adequate. The improved efficiency of the newer Pis shows through very clearly, with the new Pi 2 drawing 25% less current than the original Model B. If you're intending to run a portable station, the lithium ion battery packs that are sold to recharge phones can make an ideal 5V power pack. I have a very compact unit with a capacity of 10.4Ah that is ideal for running the Pi and keeps the Pi 2 or B+ going for more than 30 hours of continuous operation.

DATAMODES SOFTWARE. The most successful datamodes software for the Raspberry Pi is FLDIGI, as it provides a wide range of operating modes and runs surprisingly well on the Pi. I'll start by assuming you don't yet have the Linux operating system on an SD card. The type of SD card you need depends on the Pi model you have. The original B used a standard, full-size SD card, whereas models A+, B+ and Pi 2 B use micro SD cards (see **Photo 3**). Many of the micro-SD cards are supplied with

Mike Richards, G4WNC • e-mail: mike@photobyte.org

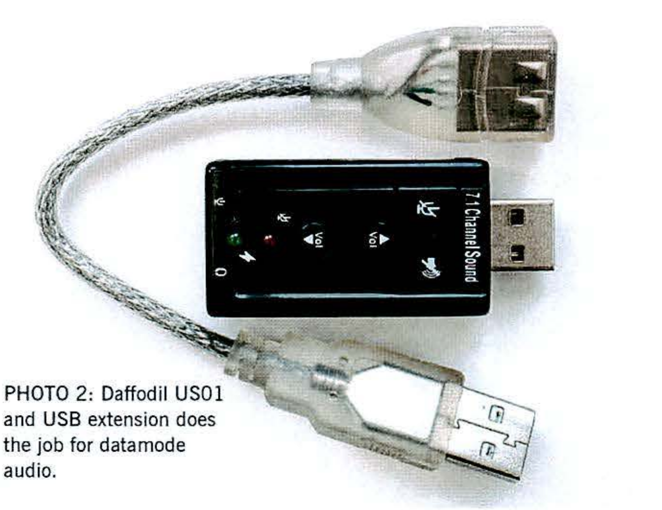

PHOTO 2: Daffodil US01 and USB extension does the job for datamode audio.

SD adapters so you can have the best of both worlds. The minimum SD card size you should use is 4GB but I suggest you buy an 8GB card as that will give you plenty of storage, plus 8GB cards are often available at a good price. I can supply ready configured and individually tested 8GB SD cards from my website: (http://photobyte.org/shop/) or you can prepare your own card. Full instructions for burning the operating system to an SD or micro SD card can be found at http://goo.gl/hKmGhe

If you're going to install the software yourself you will also need a network connection. It can be either an Ethernet cable from your router or a standard Wi-Fi dongle. The network connection is only required to download and install the software and is not required for datamode operation.

The next step is to insert the SD card into the Pi with the contacts towards the PCB, connect the network cable or Wi-Fi dongle and the sound card, then power it up. The first time you do this you will be presented with the Pi's configuration screen where I suggest you activate the following options:

Option 1. Expand the Filesystem. Select this option as it will give you access to all the storage on your SD card.

Option 3. Enable boot to desktop. Select this option and choose Desktop login as Pi. This will make the Pi boot to the graphical desktop instead of the command prompt.

Option 8. Advanced Options. Select A4 SSH. This will run the SSH server so you can access the Pi 'Headless' from your main PC as described last month. Although not essential, it's a useful facility to have.

When you've completed these entries, tab down to <Finish> and follow the instruction to re-boot the Pi.

You now have the operating system installed and configured but if you are using a Wi-Fi dongle, that also needs to be setup. This is another simple process:

1 – Choose Menu – Preferences – Wi-Fi Configuration. This will start the wpa_gui panel.
2 – Select the Adapter, which will probably be wlan0 if you have a single Wi-Fi dongle installed. If you don't see your dongle listed, just remove the dongle and re-insert it.
3 – Press Scan to open the Scan panel and press Scan again to start the search for local Wi-Fi networks.
4 – Double-click on the entry for your Wi-Fi network. This will open the configuration box where you can then enter your Wi-Fi password in the PSK box.
5 – Press the Add button and the connection will be setup and remembered.
6 – Close the Scan results panel.

NB: If you re-boot the Pi, you will need to open the wpa_gui panel again as it doesn't automatically re-connect.

Now you're ready to install FLDIGI. This is a straightforward process and the first step is to open a terminal so we can enter some command-line instructions. On the current Pi software you can find the LXTerminal in the icons at the top of the screen. When this opens, you need to type: 'sudo apt-get install fldigi' and answer yes to the prompt. The Pi will then download FLDIGI along with any other dependent software. When this completes, you can close the LXTerminal and navigate to the Menu button on the top left of your screen where you should see a new Hamradio entry. Click on this and you will see the FLDIGI entry that you can also click to start the software. As this is the first start, FLDIGI will guide you through some setup questions and you should just follow the instructions and enter your station details. When you get to the audio setup, tick the PortAudio box and select your USB soundcard from the Capture and Playback drop-down menus and click the Save button.

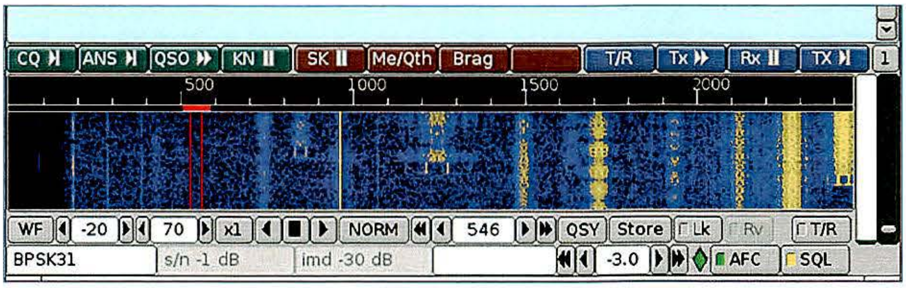

FIGURE 2: FLDIGI waterfall display with the correct input levels.

CONFIGURING FLDIGI. The FLDIGI configuration is linked to the model Pi you have, as shown here:

Pi Model A, A+, B, B+ configuration: Due to the limited processing capabilities of these models, we need to disable the multichannel detector and make sure the software is set for a slower CPU. Here's how to make those changes in FLDIGI:

1 - Choose the Configure menu then Modems – PSK – General.
2 – In the bottom section of the General tab, un-tick Multi-channel detector and press Save.
3 – Move to the Misc tab and tick the Slow CPU box and press Save.

PHOTO 3: Standard (left) and micro SD cards, with a penny and ruler marked in mm for scale.

Pi 2 Model B configuration: The additional CPU power of the Pi 2 means we can enable the multichannel detector and un-tick the slow CPU box so it's the reverse of the process for the other models.

That completes the configuration, so it's time to feed some audio from your rig to the microphone input of the USB soundcard and check all is well. Tune your rig to 14.070MHz USB and you should find some PSK activity. On the FLDIGI menu set the Op Mode to PSK-31 and you should be in business. When the receive levels are correct the waterfall display should show a blue background with yellow stripes as in **Figure 2**. If the display is predominantly yellow or red then you need to reduce the audio level. This adjustment can be done externally if you're using a data interface or you could insert a simple level potentiometer in the lead between the rig and the Pi. An alternative is to install the excellent QASmixer on the Pi (**Figure 3**), as this gives you on-screen level controls for both the input and output. To install QASmixer, open an

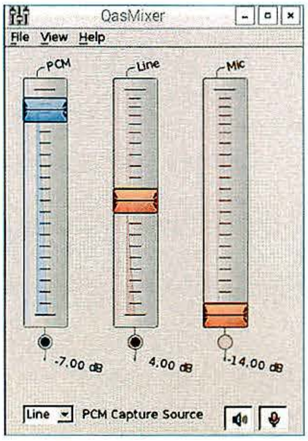

FIGURE 3: QASmixer for managing the transmit and receive levels.

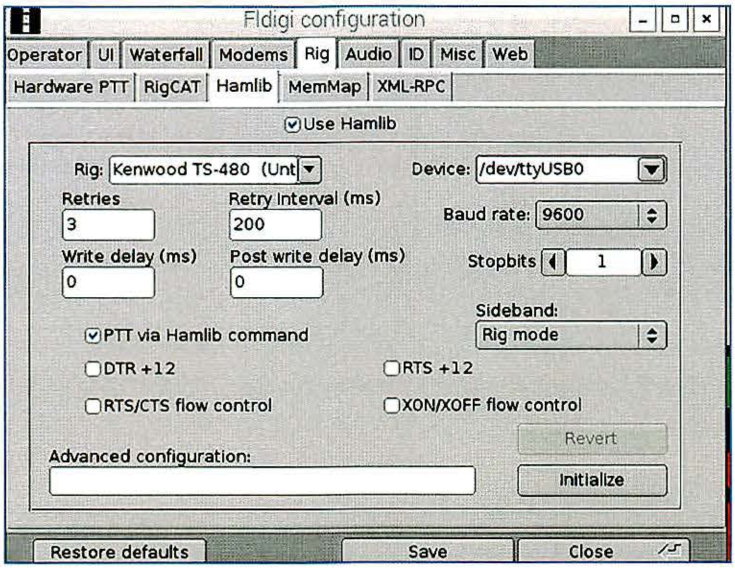

FIGURE 4: FLDIGI setting for CAT control using a USB serial connection.

LXTerminal window then type 'sudo apt-get install qasmixer' and follow the prompts. When complete, the mixer will be in the Sound & Video option in the Pi's Menu.

Transmit audio level is critical, especially for PSK, as it's important to keep the PA in linear mode. FLDIGI provides easy access to transmit level adjustment via an attenuator on the bottom line of the display. The attenuator has a 30dB range, which should meet most requirements.

FLDIGI LIMITATIONS. When running on all models except the Pi 2 you should be able to operate all modes except RTTY. For some reason, RTTY reception with FLDIGI demands more processing capacity than the older Pis can manage. However, the new Pi 2 can handle all modes including RTTY with ease. In my tests, I was able to run PSK along with the multi-channel signal browser – the Pi 2's processer was cruising at just under 25%.

RIG CONTROL. This simplest form of rig control for datamodes is to set your rig for USB with VOX enabled. Whenever your data modes software switches to transmit, the transmit tone will trigger the VOX and switch your rig to transmit. However, FLDIGI includes more sophisticated rig control and can send CAT commands via a COM port connection so you can tune the rig from FLDIGI. There are a few ways to make the serial connection to the rig but I will show you how the use a standard USB serial connector, similar to that used in a number of rig interfaces. In my shack I've got an SB2000 interface that includes a USB serial interface but many others will work just as well. Linux is a very well sorted operating system and handles plug-and play seamlessly. The first step is to plug in your serial USB lead before running FLDIGI. This is to ensure the device has been registered on the system before you try and use it. The next step is to run FLDIGI, open the Configure menu and choose Rig – Hamlib (see **Figure 4**). Here you should tick 'Use hamlib' and then choose your rig from the drop-down and set the Device box to your USB serial connection that will almost certainly be /dev/ttyUSB0. If you don't see a /dev/ttyUSB option, try restarting FLDIGI or even restarting the Pi. Next set the Baud rate and stop bits to match your rig and tick the PTT via Hamlib box. When you've completed the settings, press the Initialize button and FLDIGI will communicate with the rig and you should see the current frequency pop-up in FLDIGI. Don't forget to press Save before you leave this section or your settings will be forgotten! You should now be able to control your rig from FLDIGI.

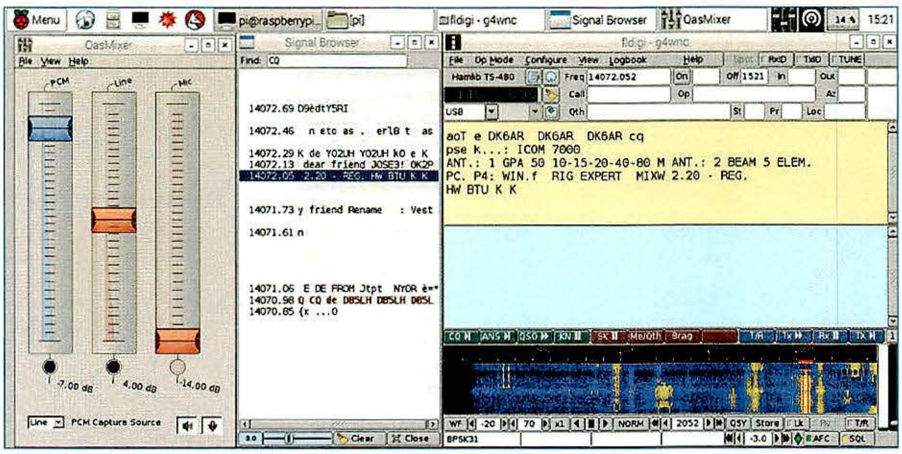

FIGURE 5: Ultimate QRP? Elad FDM-DUO driven by a Pi 2 with FLDIGI multi-channel decoding.

TABLE 1: Pi current consumption.

Pi Model	FLDIGI no network	FLDIGI with Ethernet	FLDIGI with Wi-Fi
A	460mA	630mA	580mA
A+	340mA	580mA	520mA
B	400mA	480mA	840mA
B+	250mA	300mA	400mA
Pi 2 Model B	300mA	350mA	480mA

Measurement Notes: The Model A and B tests used a 4-port USB hub that was powered by the Pi. In addition, the Model A and A+ Ethernet measurements were made using a USB to Ethernet converter that was also powered by the Pi.

ELAD DUO AND PI 2. As I had the excellent Elad Duo in for review whilst writing this, I couldn't resist linking the new Pi 2 to the Elad. I can tell you it's an ideal match! The Elad's USB CAT and USB audio leads connect directly to the Pi 2's USB ports with no problems. I was able to very quickly set up a compact station with CAT control from FLDIGI and the QAS mixer controlling the transmit and receive levels. Even with the multi-channel decoder running, the Pi 2 was cruising with the processor remaining below 25%! All you need to add is a keyboard and screen and you have a wonderfully compact multi-mode station. A sample session is shown in **Figure 5**.

SUMMARY. Hopefully, this article will have given you the guidance you need to work the world on data modes with the Pi. Clearly, the new Pi 2 will have a dramatic impact on amateur radio software and I'm quite taken by the potential to create a very attractive hill-topping rig with the Elad FDM-Duo and a PI 2. If you do get in a muddle or find loading Pi software daunting, I have pre-configured SD cards available from my website (g4wnc.com/shop).

April 2015 • RadCom
Mike Richards, G4WNC • e-mail: mike@photobyte.org

The Raspberry Pi in your shack part 3

An improvement to the Raspberry Pi dongle server and a shack security system form the final part of this series

Originally Presented at
RSGB 2014 Convention
See You in 2015

RTL DONGLE SERVER – RASPBIAN UPDATE. In the first article, I described how to use the Pidora operating system to get the RTL dongle server working on the Pi. This was necessary because the server doesn't work properly with the current Raspbian build. Since then, I've been investigating the server problem and narrowed it down to libusb-1.0. Libusb-1.0 is a code library that supports Linux USB operations. Further investigation revealed that the current Raspbian operating system is using an older version (11.1) of the library. I found that updating to libusb-1.0 version 19.1, solved the server problem. In order to force the update to version 19.1, we need to obtain the updated files from the repository used for Raspbian's testing release, known as Jessie. To implement the change, I suggest you start the upgrade process with a freshly installed Raspbian image and follow these steps:

Open a terminal session.

- Type *sudo nano /etc/apt.sources.list* to open the apt sources file.
- Add this single line: *deb http://archive.raspbian.org/raspbian jessie main*
- Hit *ctl-x* followed by *Y* then *Return* to save your changes.
- Update the cache by typing: *sudo apt-get update*
- Type: *sudo apt-get install –y libusb-1.0* this will download and install the new version.

That completes the libusb update and we can now continue with the rest of the RTL dongle server installation by following the commands in Table 1.

FIGURE 1: MotionPie web interface.

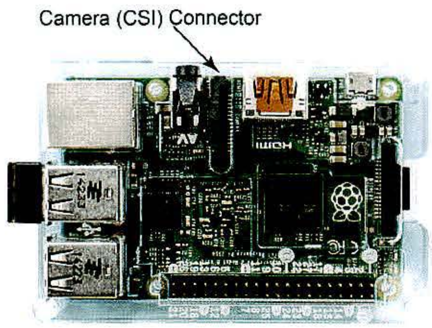

PHOTO 1: Raspberry Pi CSI Camera Connector.

AUTO-STARTING THE SERVER. One of the best ways to use the Pi dongle server is to locate it as close as possible to the antenna, thus cutting out the antenna feeder losses. The snag with this is that you have to remotely access the Pi to restart the server in the event of a power loss. The solution is to configure the Pi to automatically start the server whenever it reboots. Linux has a very simple way to do this, using what's known as the rc.local service. This service runs at boot time and will execute any commands that are stored in a file named rc.local. This file is located in the /etc/ directory and can be opened for editing by starting a terminal session and typing: *sudo nano /etc/rc.local*. On the current Raspbian build, you will see that there is already some code in that file. This is used to check for an IP address at boot and print the details to the terminal screen. If you recall, we need to know the Pi's IP address to start the RTL server so we can hack the existing script to get the IP address for our command at the same time. I've shown the modified rc.local script in Table 2. There are two additions. The first is the *sleep 10* command that provides a 10 second pause to allow a Wi-Fi connection to complete. If you're using an Ethernet connection to the Pi that line can be omitted. The second change is to add the new line: *usr/local/bin/rtl_tcp –a $_IP &*. This is the command that starts the dongle server. You will see that I've used the full path to the software and included an ampersand at the end. The full path is always recommended for local.rc scripts and the ampersand signifies that the script will continue to run in the background. The '$_IP' part of the line is the variable that contains the Pi's IP address.

You can use this same process to start any other programs at boot. If you find some of the commands a bit daunting for now, I have ready configured and tested SD cards complete with printed instructions available from my website, http://g4wnc.com.

SHACK SECURITY. With our radio shacks often located in outbuildings, security is always a concern but the Raspberry Pi can lend a hand here as well. One of the Pi's many fortes is its ability to operate as a web server to share information across your home network. One such application is to use the Pi as a camera server for a security system. One Pi can be setup to act as a hub for a number of webcams plus a Raspberry Pi camera. In a typical installation you could have one camera located inside the shack and perhaps one or two more overlooking the outside. The Pi can be set to stream these images over your local network so you can take a look any time you like using your PC, smartphone or tablet. You can also set the Pi to automatically start a video recording when any movement is detected. I have set this up previously using an article I found at http://goo.gl/MEH5MY. However, I've recently found a much easier solution, thanks to some excellent work by Calin Crisan. At the heart of the solution is the Motion software that's been around for a while on Linux. Motion is the workhorse that can manage images from multiple cameras and use them to detect movement, trigger recordings or raise alarms. Whilst the software is very good, it has a huge and rather unwieldy configuration system that's often a step too far for those that try it.

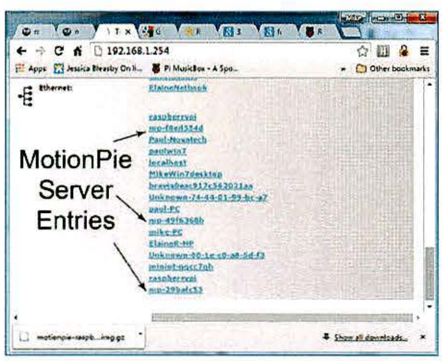

FIGURE 2: Home Router web interface showing multiple MotionPie servers.

Mike Richards, G4WNC • e-mail: mike@photobyte.org

FIGURE 3: MotionPie general settings.

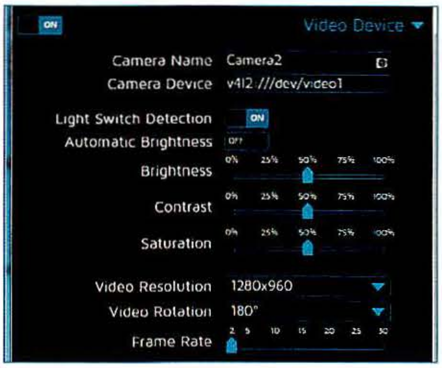

FIGURE 4: MotionPie video settings.

FIGURE 5: MotionPie motion settings.

Whilst I have worked through it successfully in the past, it's certainly not a job for the Linux newbie. What Calin has done so well, is to create a very easy to use web interface to Motion that makes it a delight to use. As if that wasn't enough, he has also created a dedicated SD card image. The image includes a minimal Linux operating system, effectively creating a plug and play implementation. I've been in contact with Calin and he's happy for me to make pre-programmed SD cards available to *RadCom* readers. See my website for details.

If you want to burn your own SD card, the software is completely free of charge and can be downloaded from: http://goo.gl/iclyJF. This is a complete, ready to go, operating system and application that's been built using Buildroot (http://buildroot.uclibc.org/). Be careful to download the correct image, as there are separate images for the Pi 1 and Pi 2. Once you have the image, all you have to do is unzip it, write it to an SD card and boot the Pi. If you prefer to use a ready prepared SD card, I have these available from my website, http://g4wnc.com. Each SD card is individually tested, ready to go and comes with comprehensive printed instructions.

INSTALLATION AND USE. If you want to download and burn your own SD card, here's a run through the process using a Windows PC to prepare the image. The MotionPie image is supplied as an .img.gz compressed file so the first step is to expand it. One of the best and most popular, free, tools for this task is 7-Zip as it handles all the common file compression systems. Download and install 7-Zip and use it to expand the MotionPie file. At the end of the process you should have a file with an .img suffix. The next task is to write that image to your SD card. To do that you need two more free tools. SD Formatter is a utility specially designed to format SD cards and can be downloaded from http://goo.gl/OcnYdZ. The final tool is Win32DiskImager, available from http://goo.gl/n5H2C6.

Here's the SD card creation process in detail:

- Plug your SD card into your computer using a suitable card reader.
- Run SD Formatter and format your new SD card. Please take extra care to ensure you choose the correct card or you could wipe one of your other disks!
- Close SD Formatter and run Win32DiskImager.
- Select the drive letter used by your SD card for the Device field.
- Click the folder icon and navigate to the .img file that you downloaded and expanded.
- *Double-check* the settings and click Write to transfer the .img to the SD card.

With the SD card prepared, you can pop the card into your Pi, connect an Ethernet cable and power it up. There's no need to connect a keyboard, screen or mouse as the MotionPie will boot on its own and is fully accessible via its web interface.

TABLE 1: Commands required to complete the installation of the RTL dongle server.

Open a terminal session.
Type *sudo apt-get install –y cmake pkg-config*
Download the software from the Osmocom Github site: *git clone git://git.osmocom.org/rtl-sdr.git*
Type: *cd rtl-sdr/*
Type: *mkdir build*
Type: *cd build*
Type: *cmake ../ -DINSTALL_UDEV_RULES=ON*
Type: *make*
Type: *sudo make install*
Type: *sudo ldconfig*
To start the server, type: *rtl_tcp –a* 'your Pi IP address'

CONNECTING UP MOTIONPIE. The first step is to connect a camera or two and there are various options. An obvious choice is to use one of the excellent Raspberry Pi dedicated camera models, of which there are two choices. The standard camera is a good quality 5 megapixel unit, whilst the Pi Noir is the same camera but with the infrared filter removed. This makes the Noir version sensitive to infrared light, which is better for night-time security surveillance. I have used both cameras successfully but you can only use one on a Pi because it connects using the dedicated CSI (Camera Serial Interface) socket on the PCB, see **Photo 1**. However, you can combine a Pi camera with readily available USB connected webcams that are available from all the usual online suppliers. If using the original Pi 1, you will have to keep a close watch on the power consumption as the Pi 1 has limited power available for USB devices. This limitation is due to the use of a 750mA polyfuse in the incoming 5V supply line. If you start getting erratic behaviour, it's a common sign of power problems. The simplest solution for multiple webcams on a Pi 1 is to use a powered USB hub. The new Pi 2 is much better behaved as it has four USB ports and can handle a 2A total consumption. On my system, a Pi 2 running MotionPie with the Pi camera connected used around 500mA and connecting a Logitech 9000 webcam raised that to 700mA. I suggest you use a 5 volt, 2 amp supply if you're connecting several cameras to the Pi 2.

CONFIGURING MOTIONPIE. Once you have your Pi up and running with a camera or two connected, it's time to access the web interface, see **Figure 1**. To do this, you need to know the IP address that MotionPie was allocated when it connected to your network. The simplest way to find the IP address is to log in to your home router and find the home network page where you should see the MotionPie listed as one of the connected devices, see **Figure 2**. The network name given to the MotionPie is 'mp' followed by the Pi's serial number. On my system the MotionPie shows as 'mp-29bafc53'. Once you've located the MotionPie on your router, it's a good time to configure the router to allocate the same IP address. This is a neater solution than

Mike Richards, G4WNC • e-mail: mike@photobyte.org

setting up fixed IP addresses. If you have a smartphone you can also use the Fing app to search your local network and locate the MotionPie's IP address.

All the MotionPie configuration is handled via the excellent motionEye web interface. To access this from any PC, tablet or smartphone just type the MotionPie's IP address into your browser. This will open up the interface as shown in Figure 1. To login, use 'admin' as the username with no password. At the top left of the screen you will see the settings icon that looks like three sliders – click on that. Starting with General Settings at the top, make sure this is set to ON and click the arrow to open up the settings, see **Figure 3**. Here you can make the system more secure by setting new usernames and passwords to control access. The surveillance name/password just allows people to view the cameras/video whereas the administrator has full access to all the controls and can delete videos. Don't forget to make a note of any password/name changes as they cannot be recovered!

Next we need to configure the cameras on the system. To do that skip down to the Video Device and click the arrow to open up the options, see **Figure 4**. As part of the boot process, MotionPie automatically includes any cameras that are connected to the Pi. To switch between cameras or to add a new camera use the camera selector drop-down at the top of the panel. The Video Device settings contains lots of useful adjustments so you can get the best from your setup. If you are using multiple cameras you may want to give them more appropriate names: that's done using the Camera Names field. You will also see adjustments for brightness, contrast and saturation. These are particularly useful for some of the cheaper webcams. Next is the Video Resolution and you can select from a range of resolutions. The trick here is to select the lowest resolution that gives a useable image quality. Higher resolutions result in larger file sizes so you will use up storage space more quickly or generate more network traffic when accessing the images. On my system, I have the Raspbery Pi Camera setup for 1280 x 960 resolution whilst my webcams use 960 x 720. The Video Rotation setting is very useful as you will often find that the only suitable fixing leaves the camera on its side or even upside-down. The Video Rotation lets you straighten up the image. Finally, in this section is the Frame Rate. For security surveillance, the default 2 frames per second is about right and helps keep the video file size down. The next essential adjustment is the File Storage, this is where MotionPie will save its video files. You can use the Pi's SD card but that will quickly run out of space with large video files. There are a few alternatives such as a USB memory stick, USB portable hard drive or some form of network storage. Memory sticks are easy to use and 32GB sticks can be had for under £10. The MotionPie software will automatically recognise USB sticks and external drives and they will be available via the Storage Device drop-down menu. If your stick isn't recognised, try rebooting the Pi. I'm using one of the very compact Samsung portable drives on my system but I found that it needed a powered hub, even with a Pi 2.

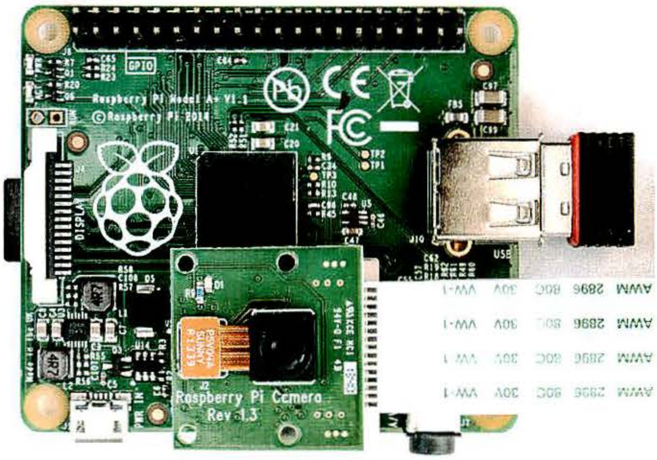

PHOTO 2: Compact Wi-Fi server with Raspberry Pi model A+ and camera.

Moving down the list of options, make sure that Text Overlay and video streaming are both on and move down to Motion Detection, see **Figure 5**. This is where you can control the movement detection system. The 'Frame Change Threshold' determines what percentage of the image needs to change to trigger a recording. This will depend on the type of scene you are monitoring. If your camera is set up for a wide view of your drive, then a person walking up the drive will only represent a small change in the overall image so you will need a small setting. However, if you have a camera on the front door, the potential intruder will be much larger in the frame so you can set a higher threshold. The optimum setting is usually found by trial and error to give the best compromise between reliable detection and spurious triggers. The remaining video settings can be left at their default values. In the Motion Movies section you can adjust the movie file names, set the movie quality and most importantly, decide how long movie files are retained. This is a very useful way to keep your storage under control as MotionPie will automatically overwrite older files when it starts to run out of space. If you're feeling particularly paranoid, the notification setting lets you setup email or other alerts when movement is detected!

TABLE 2: modified rc.local script (see text).

```
sleep 10
_IP=$(hostname -I) || true
if ["$_IP"]; then
    Printf"My IP address is %s\n" "$_IP"
    usr/local/bin/rtl_tcp -a $_IP &
fi
exit 0
```

You can use the Woking Schedule panel to set the activation times. This is ideal for home situations where you want it to kick in overnight rather than recoding your daily routine.

WIRELESS ACCESS. Whist I recommend you use an Ethernet connection to the Pi to give decent network streaming, MotionPie can also be connected via Wi-Fi. The easiest way to configure the Wi-Fi is to start your Pi with an Ethernet connection and login as described earlier. Once you're logged into the MotionPie interface, go to the Network settings and type in the SSID and network key for your Wi-Fi system and press Apply. That's all there is to it and you can remove the network connection, plug in the Wi-Fi dongle and reboot the Pi. I've used this technique to create a very compact Wi-Fi combination with a Pi Model A+, Wi-Fi dongle and the Pi Camera, see **Photo 2**. My plan is to mount this inside a dummy security camera.

SUMMARY. That completes this short series on the wonderful Raspberry Pi and I hope it has encouraged you to get that Pi out of the drawer and get it doing something useful. The recent introduction of the Pi 2 is a major improvement that opens up the Pi to tasks we couldn't consider with the previous version. Work is currently in hand to get Joe Taylor's WSJT and WSPR suite working on the Pi 2 as well as GNUradio. The combination of the graphic processor's FFT (Fast Fourier Transform) library and the faster Pi 2 processor should also spawn some practical SDR designs. If you're new to Linux and want to get started quickly, I have pre-loaded SD cards for all the software covered in this series. Each card is individually tested and supplied with comprehensive printed instructions. These cards can be ordered via my website (http://g4wnc.com).