

Building a Shelter Information Distribution System Independent of Normal Telecommunications Systems



by Gordon Gibby

Original document: 2018

Updated for BUSTER Raspberry Pi Operating System and Raspberry Pi 4, February 2021

NOTE: Throughout this educational document, port numbers and passwords have been given dummy assignments which are NOT the same as those in deployed units, for obvious security reasons.

After action review of the Alachua County response to Hurricane Irma revealed a significant need for methods to get information out to hundreds, or thousands of county shelter residents. With several hundred people to a shelter, and running 13 shelters, it proved difficult to keep information flowing --- and almost impossible if hurricane-force winds were to take out grid power, cell towers, internet and other resources. Shelter residents overflowed available middle- and high-school classrooms, and were impossible to bring together for announcements or information.

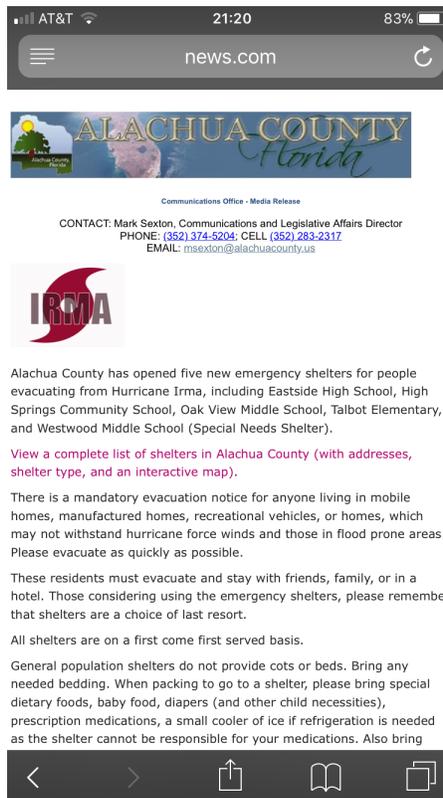
Yet these days, almost every person in a temporary shelter has *a working cell phone* – a SCREEN on which information can be viewed. The problem is how to get information to those screens if normal cell-service systems quit. **Hint: they still get WIFI!**

I concluded that a workable solution would be to use an inexpensive Raspberry-Pi-based Web Server, driving a standard home consumer WIFI router. This could be given the SSID of “Shelter-A” etc., operate from storage batteries, and be pre-loaded with all manner of useful background information

(such as shelter rules, forms, ham radio information, FEMA contacts, assistance explanations) --- and get urgent storm updates and other essential information via 2-meter packet and/or WINLINK through shelter ham radio volunteers. The hams would use either .bat file or graphical FTP techniques to upload new html content. Inexperienced hams could even find detailed instructions pre-stored on these miniature web servers.

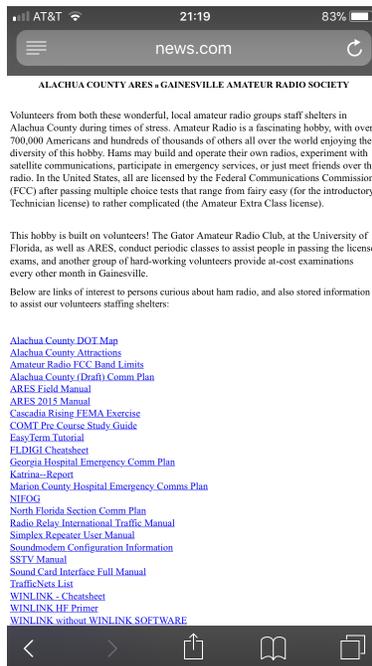


Opening Screen Users might see when accessing our standalone web system on a cell phone.



Hurricane Update users might find on a standalone shelter information system driven by ham radio.

Graphics, maps, normal schedules, ICS forms, instruction sheets, first aid information – almost an encyclopedia – could be preloaded onto to the huge micro-SD card of the Raspberry. County EOC headers and other graphics that are customary (and unchanging) in their press releases could be preloaded as well. Then simple text-type .html files could allow them to have an up-to-date current status always available on the shelter backup web server for viewing by residents.



*Ham radio operators might have access to a plethora of stored files
– frequency charts – instruction manuals and other valuable sources.*

Shelter hams could receive error-free updates to the text .html file using any of a number of methods;

- Winlink email attachments
- Easyterm packet YAPP file transfer protocol
- NBEMS

The file EOC.html is the placeholder for the beginnings of the County-produced information, whatever they wish. Doing this as .html instead of .pdf reduces the size of the file that might have to be transferred if there is no other method of receiving updates, than radio.

I can easily copy a microSD card for ARES (R) or other groups who wish to try this out and have lesser familiarity with Linux and Raspberry pi's.

An explanation of how the networking for this system works:

The Raspberry Pi carries out the functions of an Internet Service Provider:

- It establishes its own IP number as 10.10.10.10
- It provides DNS services to downstream devices and associates the names hamradioband.com and www.hamradioband.com with the same IP number (its own) 10.10.10.10

- When it detects that a wifi router has made an ethernet cable connection to it it provides it a new IP number on its ethernet connection, in the 10.10.10.X network. It also provides DNS services to the router. More than one router can be connected (using a switch) to the Raspberry and all will be treated appropriately.

Each consumer wifi router than fields a WIFI-based network, on a network such as 192.168.1.X, providing network address translation to the IP number that it received from the Raspbery, and thus passings requests from users to the Raspberry pi for pages to be served.

It is suggested that the consumer wifi routers be set up as follows;

1. Provide a password on administrative changes to the router that only you know.
2. Make the router provide a distinctive SSID such as “SHELTER INFO SERVER” or similar.
3. Make the router allow users to connect without any passphrase or password.

An example list of the items that might be stored on the HAM RADIO page on such a micro-server is:

[Alachua County DOT Map](#)
[Alachua County Attractions](#)
[Amateur Radio FCC Band Limits](#)
[Alachua County \(Draft\) Comm Plan](#)
[ARES Field Manual](#)
[ARES 2015 Manual](#)
[ARRL RADIOGRAM](#)
[Cascadia Rising FEMA Exercise](#)
[COMT Pre Course Study Guide](#)
[EasyTerm Tutorial](#)
[FLDIGI Cheatsheet](#)
[Katrina--Report](#)
[NIFOG](#)
[North Florida Section Comm Plan](#)
[Radio Relay International Traffic Manual](#)
[Simplex Repeater User Manual](#)
[Soundmodem Configuration Information](#)
[SSTV Manual](#)
[Sound Card Interface Full Manual](#)
[TrafficNets List](#)
[WINLINK - Cheatsheet](#)
[WINLINK HF Primer](#)
[WINLINK without WINLINK SOFTWARE](#)

[WINLINK CheckLists](#)
[WINLINK - How PACLINK works](#)

INCIDENT COMMAND SYSTEM BLANK FORMS

[ICS - 213 MESSAGE FORM](#)

[ICS214 - ACTIVITY LOG](#)

[ICS309 - COMM LOG](#)

As mentioned earlier, I'll be happy to build you a chip for your Raspberry version 3 if you contact me and send me a blank chip. You can then copy the chip as you need. Or you can build the entire Raspberry Pi system yourself in a matter of an hour. However, that takes some familiarity with linux and with TCP/IP. If I build you a chip, it will come with a table of the usernames / passwords and ports similar to the following:

Username	Password
pi	xyz
root	prstuv!!
FTP User: (for uploading data files)	
ARESHAM	bossham

SERVICE	Port
sshd	8078

(The actual passwords and ports are different from those tables, and will be provided to actual recipients.)

BUILDING IT YOURSELF

The details of the technical development of this system are as follows:

Start with a standard Raspberry Pi Version 4 with Raspberry pi "buster" version of the linux operating

system installed. You'll need a few hundred megabytes of free space at a minimum; I recommend a 16Gbyte microSD card, although the system will fit into an 8Gbyte one. If the system is going to operate continuously, I recommend that you get a microSD card built for continuous filming, such as “dashcam” use.

INITIAL SECURITY

- CHANGE the default password of user pi, to be something unique and write it down so you don't lose it.
- CHANGE the default password of user root some be something unique and fairly secure, and write it down so you don't lose it!!!! *If you lose that password you can use the pi user and sudo to change the root password, with a little ingenuity.*

UPDATE SOFTWARE

Now open a terminal window as user pi, and execute the following to update your system's package list.

```
sudo apt-get update
```

CREATING HIDDEN SECURE SHELL ACCESS

You will want to have secure shell access so that you can remotely control the Raspberry pi using a secure connection (“secure shell”) sshd The explanation of how to do this can be read here: <https://www.raspberrypi.org/documentation/remote-access/ssh/>

I would strongly suggest that you move the sshd access port from its well-known port of 22, to a different port, such as 9512. Otherwise, your system, if ever left connected to the internet for any length of time, will be attacked by “bots” who will try passwords forever in an attempt to enter your system. You change the port in this file:

```
/etc/ssh/sshd_config
```

(Look for the line #Port 22, remove the # and change the port number.) When you install your firewall (ufw, see later in this document), be certain that you provide sshd access through the proper port number – has to be the same one you put in sshd_config! Don't use port 9512 (which we use in this document as an example) – use something in the 8000's other than 8080, and be assured that my prototype system doesn't use 9512 either!

Suggestion for terminal access from a windows computer:

I use the free package PuTTY to control Raspberry Pi systems using ssh:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

INSTALLING THE APACHE WEB SERVER

The following URL give information on installing the Apache web server:

<https://maker.pro/projects/raspberry-pi/raspberry-pi-web-server> My instructions are close to theirs.

```
sudo apt-get install apache2 apache2-doc apache2-utils
```

SKIP THIS ON RASPBERRY PI BUSTER 3B+ / 4 INSTALLS

(Note: the following line doesn't seem to work and is no longer recommended:)

```
sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache
```

Your web files will go in the subdirectory `/var/www/html` --- and you can add additional subdirectories below that to better organize your files, but that is probably overkill. Be certain that you eventually end up with an `index.html` in the `/var/www/html` directory.

PROVIDING DNS & DHCP SERVICES TO ROUTER & USERS

Your raspberry pi must provide DNS and DHCP services to the standard home wifi router that connects to it, and expects it to “act like an Internet Service Provider.” The URL that has the relevant background in instructions is: <http://www.raspberrypi.org/learning/networking-lessons/lesson-4/plan/>

```
sudo apt-get install dnsmasq
```

REBOOT THE RASPBERRY AT THIS POINT. THIS MAY BE SUPERFLUOUS BUT MAY AVOID UNEXPECTED PROBLEMS.

AFTER REBOOTING, Become root (sudo root, and provide the proper password) – then

**SKIP THESE BOXED STEPS ON RASPBERRY BUSTER 3B+ / 4 INSTALLATIONS
THEY FOUL UP THE WIFI OF THE RASPBERRY & MORE -- THEY ARE APPARENTLY
ERRONEOUS**

edit `/etc/network/interfaces`, comment out (with a # at the first of the line) the line

```
iface eth0 inet manual
```

and add these lines after, so it shows:

```
#iface eth0 inet manual
auto eth0
iface eth0 inet static
address 10.10.10.10
netmask 255.255.255.0
```

Then add the following lines to `/etc/dhcpd.conf` (note carefully that file's spelling!--there are two c's)

```
interface eth0
static ip_address=10.10.10.10/24
static domain_name_servers=10.10.10.10
```

As root, edit file `/etc/dnsmasq.conf` (which just have scores of commented out line beginning with #) and add the following to the end:

```
interface=eth0
dhcp-range=10.0.0.100, 10.0.0.254, 255.255.255.0, 24h
dhcp-option=6,10.10.10.10
no-hosts
addn-hosts=/etc/hosts.dnsmasq
```

These tell dnsmasq to provide dhcp-provided IP numbers on interface eth0 ranging from 10.10.10.100 to 10.10.10.254 and lease them for up to 24hours. (You can now handle more than 100 home routers, but you probably only need one or two to handle an entire shelter full of residents.)

It tells it to provide nameserver services from 10.10.10.10 and notify routers getting connection from it of that fact, and that it will not look at the `/etc/hosts` file, but instead the `/etc/hosts.dnsmasq` file for known names.

So now we need to create the `/etc/hosts.dnsmasq` file! As root, edit file `/etc/hosts.dnsmasq` to have a couple lines like this:

```
10.10.10.10    hamradioband.com
10.10.10.10    www.hamradioband.com
```

NOTE: to guarantee that I wouldn't cause problems if anyone accidentally connected this system to the real internet....I purchased the domain name hamradioband.com for 2018 and 2019. Any ARES group is welcome to use that domain name in this system....it isn't mapped anywhere else...

IN GENERAL -- you can advise shelter residents to point their browser to 10.10.10.10 and it works fine. You don't have to use these DNS names at all.

REBOOT

Now your system will come up, respond to the name www.hamradioband.com and the ip number 10.10.10.10, provide an IP number in the 10.10.10.x net to router(s) that connect to it (you can power more than one router from this!) and provide web pages.

Check some of this by executing the command `ifconfig`

- It should show eth0 having IP number 10.10.10.10
- You should be able, as root, to ping hamradioband.com

```
ping hamradioband.com
```

and it should show you the ip number 10.10.10.10 and replies.

Connect the raspberry's ethernet jack to the WAN plug of a suitably configured wifi router, connect to that router with your laptop(either on WIFI or with ethernet) and you should now be able to find the default page with a browser directed to either 10.10.10.10 or hamradioband.com. If that doesn't work, log into the wifi router and see whether or not it received an IP number in the 10.10.10.x range....if not, something above didn't work.

CREATING UPLOAD FTP ABILITY

Now we need a way for you to upload new pages over FTP. In this case I wish to have a user named ARESHAM have access to the base directory of the server, password bossham. (*I didn't really use that password, and you shouldn't either*)

The URL <https://www.raspberrypi.org/documentation/remote-access/ftp.md> will help you install pureftp

```
sudo apt-get install pure-ftpd
sudo groupadd ftpgroup
sudo useradd ftpuser -g ftpgroup -s /sbin/nologin -d /dev/null
```

Set up a user ARESHAM whose files will go to /home/pi/ARESHAM but show up under <http://hamradioband.com/>

```

sudo mkdir /home/pi/ARESHAM
sudo chown -R ftpuser:ftpgroup /home/pi/ARESHAM
sudo pure-pw useradd ARESHAM -u ftpuser -g ftpgroup -d
/home/pi/ARESHAM
sudo pure-pw mkdb

```

Notice that the mkdb effort has to be done after each new user is added, and that during this process, the operating system will ask you for the password for the account ARESHAM.

Now create an authentication method:

```

sudo ln -s /etc/pure-ftpd/conf/PureDB /etc/pure-ftpd/auth/60puredb

```

Now we need to deal with the /var/www/html subdirectory --- as root, delete it

```

rm -r /var/www/html

```

As root, create symbolic links that will put material uploaded into these two directories onto the Apache web server's files.

```

ln -s /home/pi/ARESHAM /var/www/html

```

(You can reboot to make all this happen or else restart the pure-ftpd service:

```

sudo service pure-ftpd restart

```

A REVIEW OF YOUR SETUP

User	purpose	password	home directory for ftp uploads	where their material shows up on the web server
ARESHAM	ham ownership of the server	bossham	/home/pi/ARESHAM	http://hamradioband.com/

HIGHER LEVEL SECURITY

From the Raspberry pi Menu system, select Raspberry Configuration, go to interfaces, and turn SSH ON rather than disabled.

As root, edit the file `/etc/hosts.allow` so that only certain types of access are allowed into your system:

```
ALL:      192.168.  
ALL:      10.10.10.  
sshd:     ALL
```

This allows you or others connected via your WIFI routers to have relatively unlimited access to your system, but if it is connected to the real internet.....ONLY the ssh access is allowed.

FIREWALL SECURITY

Now, as user pi, add a firewall to the system

```
sudo apt-get install ufw  
sudo ufw allow 9512/tcp      <<----be sure to use correct port  
sudo ufw allow 8080/tcp  
sudo ufw allow 80/tcp  
sudo ufw allow ftp  
sudo ufw allow dns  
  
sudo ufw allow 67/udp  
sudo ufw allow 68/udp  
  
sudo ufw enable
```

Note earlier in this document, we moved sshd to operate on port 9512 .

In order to check the status of the firewall,

```
sudo ufw status
```

Now to be ready to operate inside a shelter, TURN OFF THE WIFI on the Raspberry Pi!! This prevents users from simply routing through your Raspberry, and also removed the chance for hackers on the Internet to gain access to it.

That should do it! Your system should function properly!

Tests you should run:

- Connect to your wifi router and verify that you get a connection, and can ping it.
- Note that you probably cannot ping 10.10.10.10
- If your wifi router has an administration page, you may find that it has received an IP number from the Raspberry, but that it has recognized that it isn't really connected to the internet.
- You should be able to gain ssh access to 10.10.10.10 (or hamradioband.com) on the port that you've selected for that use.
- You should be able to ftp into hamradioband.com as user ARESHAM, if you provide the proper password. Be certain to upload a proper index.html file! Material you upload should be visible on <http://hamradioband.com/>

Using a laptop computer connected to your consumer WIFI router, you should now be able to ftp content onto either user's sections of the web site.

Using the same laptop, you should be able to retrieve the content using any standard web browser.

WIFI ROUTER SETUP

Your wifi router should be set up like a normal household router, thinking it is connected to a WAN IP system, with DHCP providing it an IP number, and with a password on its administrative access, so that a hacker can't foul it all up inside the shelter. However, do NOT put a password or passphrase on the WIFI connection – there's nothing secure here so not need for security on the transmissions themselves, and it will only complicate making this asset available to shelter residents.

You want to have to tell them ONLY that they need to connect to this WIFI SSID (e.g. SHELTER INFO SERVER) and point to 10.10.10.10 with their browser.

The wifi router will likely "complain" that it can't find the Internet and it may make continuous repeated tries to find it (unsuccessfully).

The Router should be set to automatically acquire an IP address using DHCP from the “internet service provider” and also to be told to find where DNS services are automatically (it will be given 10.10.10.10).

I recommend that you give the router an easily recognizable SSID such as SHELTER INFO SERVER

UPLOADING CONTENT INTO THE SERVER

Material can be easily uploaded using a free ftp graphical program such as CoffeeCupFTP,

<https://www.coffeecup.com/free-ftp/>

or even with Windows 10 command-line ftp commands. You can find more information about using ftp on windows computers here:

<https://www.howtogeek.com/272176/how-to-connect-to-ftp-servers-in-windows-without-extra-software/>

If you use batch files (.bat) to upload files, you'll need two files, as the first will merely start the ftp process and refer to the 2nd for username/password entry and further details. As an example:

connectfile1.bat:

```
ftp -s:connectfile2.bat hamradioband.com
```