ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents:

Using the Raspberry PI
Digital modes

http://www.icarc.org/Raspberry_PI.htm

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI 5 Cost

Vilros: 4GB $60.00   8GB $80.00
PiShop: 4GB $60.00   8GB $80.00
SparkFun: 4GB $60.00   8GB $80.00

All about the same $$$
All about the same stock level (hens teeth)

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI 4 Cost

Vilros: 2GB $45.00

PiShop: 2GB $45.00

SparkFun: 4GB $45.00 8GB $75.00

Similar $$$

All seem to have stock

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI O/S

Download from
https://www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-64-bit
(the Raspberry-PI website)

Copy to SD card
Plug it in and turn on power

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Software

```
sudo apt-get update
sudo apt get install package
```

6
7

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Packages (loooooong list)

psk31
Wsjt

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Hardware

Lotsa junk on a little board.
Much channel sharing: ethernet controller on USB channel
PWM audio out
HDMI video
that little 40-pin header

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Channel Sharing

Raspberry-PI 5 has more dedicated channels
(less sharing bandwidth)

Raspberry-PI 4 adequate for digital modes!
Raspberry-PI 3 also adequate for digital modes!

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Serial

USB serial dongles all just work. Drivers already installed.
FTDIchip (like the FT4232 4-port)
SiLABS CP2102 (in the Japanese radios)

CP2102 driver ***hell*** caused by SiLABS trying to combat chip pirates.
FTDIchip when through similar fits, but seems to have quit more successfully

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Video

PI3B+ has comfortable native screen resolution
Configure size for convenient use with VNCVIEWER
PI sits by radio without keyboard or video connection.

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Audio

Icom SDR
Yaesu SDR
Kenwood SDR

all have PCM290x or derivative (48KHz)

Find audio device name under Linux: **aplay -l**

```
card 1: CODEC [USB Audio CODEC], device 0: USB Audio [USB Audio]
```

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Radio Configuration Utility

KC0JFQ Configuration Utility

Loads complete radio configuration to affect a mode change
(‘cause I can’t remember how to do it correctly)

ICARC Using the Raspberry-PI

Presentation by KC0JFQ

Preparing Raspberry-PI for software install

Allocate device name and address on your network

Download Latest Image

Copy to SD card

Move SD to RPI and apply power

Configure using setup screen and expand file system (note MAC address)

Update router with MAC address and device name

Download applications and Utilities

Connect to Radio and begin!

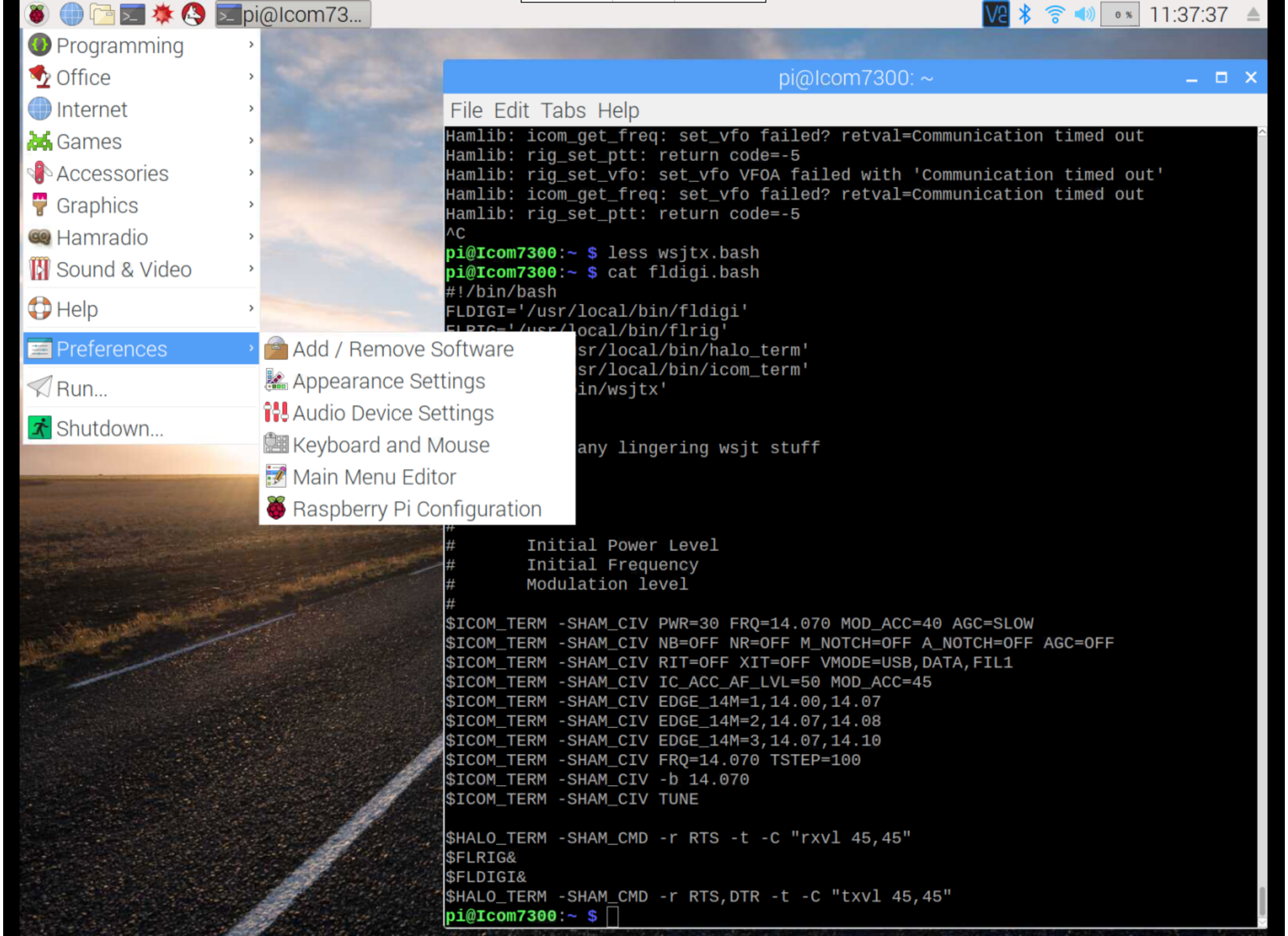
ICARC

Using the Raspberry PI

Presentation by KC0JFQ
14

Raspberry-PI Menu

Chromium Web browser slow on PI3B+



```
Hamlib: icom_get_freq: set_vfo failed? retval=Communication timed out
Hamlib: rig_set_ptt: return code=-5
Hamlib: rig_set_vfo: set_vfo VFOA failed with 'Communication timed out'
Hamlib: icom_get_freq: set_vfo failed? retval=Communication timed out
Hamlib: rig_set_ptt: return code=-5
^C
pi@Icom7300:~ $ less wsjtx.bash
pi@Icom7300:~ $ cat fldigi.bash
#!/bin/bash
FLDIGI='/usr/local/bin/fldigi'
FLRIG='/usr/local/bin/flrig'
                     sr/local/bin/halo_term'
                     sr/local/bin/icom_term'
                     in/wsjtx'


                     any lingering wsjt stuff




#
#       Initial Power Level
#       Initial Frequency
#       Modulation level
#
$ICOM_TERM -SHAM_CIV PWR=30 FRQ=14.070 MOD_ACC=40 AGC=SLOW
$ICOM_TERM -SHAM_CIV NB=OFF NR=OFF M_NOTCH=OFF A_NOTCH=OFF AGC=OFF
$ICOM_TERM -SHAM_CIV RIT=OFF XIT=OFF VMODE=USB,DATA,FIL1
$ICOM_TERM -SHAM_CIV IC_ACC_AF_LVL=50 MOD_ACC=45
$ICOM_TERM -SHAM_CIV EDGE_14M=1,14.00,14.07
$ICOM_TERM -SHAM_CIV EDGE_14M=2,14.07,14.08
$ICOM_TERM -SHAM_CIV EDGE_14M=3,14.07,14.10
$ICOM_TERM -SHAM_CIV FRQ=14.070 TSTEP=100
$ICOM_TERM -SHAM_CIV -b 14.070
$ICOM_TERM -SHAM_CIV TUNE

$HALO_TERM -SHAM_CMD -r RTS -t -C "rxvl 45,45"
$FLRIG&
$FLDIGI&
$HALO_TERM -SHAM_CMD -r RTS,DTR -t -C "txvl 45,45"
pi@Icom7300:~ $
```

ICARC

Using the Raspberry PI

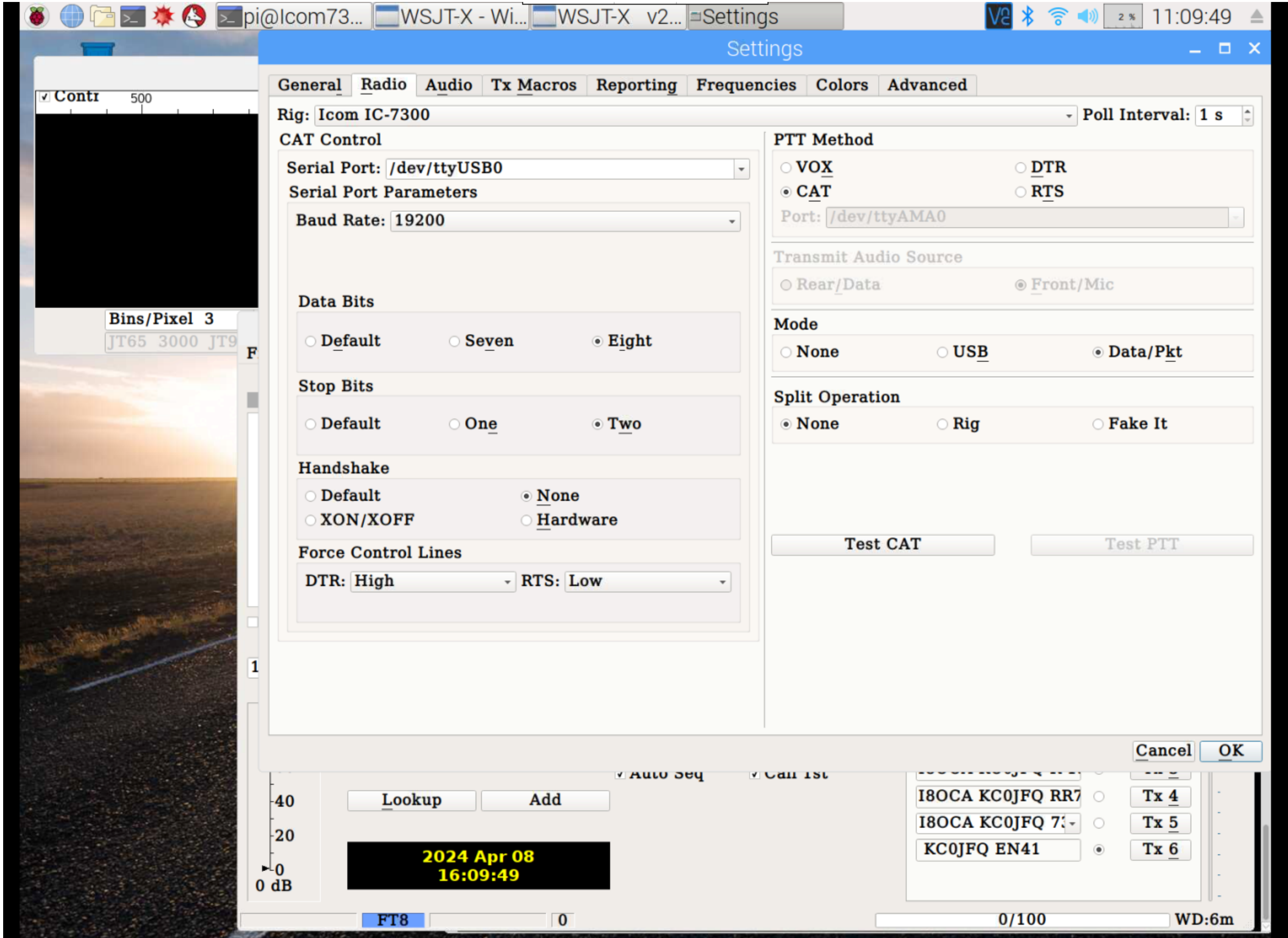
Presentation by KC0JFQ

15

WSJTX

the setup screen

CARD=CODEC,DEV=0



Settings

Rig: Icom IC-7300 Poll Interval: 1 s

CAT Control

Serial Port: /dev/ttyUSB0

Serial Port Parameters

Baud Rate: 19200

Data Bits: ○ Default ○ Seven ● Eight

Stop Bits: ○ Default ○ One ● Two

Handshake: ○ Default ● None ○ XON/XOFF ○ Hardware

Force Control Lines: DTR: High RTS: Low

PTT Method: ○ VOX ○ DTR ● CAT ○ RTS

Port: /dev/ttyAMA0

Transmit Audio Source: ○ Rear/Data ● Front/Mic

Mode: ○ None ○ USB ● Data/Pkt

Split Operation: ● None ○ Rig ○ Fake It

Finding the CODEC and Serial:

dmesg: (ICOM7300 on XEON box)

```
[12076998.001548] usb 1-1: new full-speed USB device number 25 using xhci_hcd
[12076998.127963] usb 1-1: New USB device found, idVendor=0451, idProduct=2046, bcdDevice= 1.25
[12076998.127968] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[12076998.130000] hub 1-1:1.0: USB hub found
[12076998.130117] hub 1-1:1.0: 4 ports detected
[12076998.403538] usb 1-1.1: new full-speed USB device number 26 using xhci_hcd
[12076998.481264] usb 1-1.1: New USB device found, idVendor=10c4, idProduct=ea60, bcdDevice= 1.00
[12076998.481269] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[12076998.481273] usb 1-1.1: Product: CP2102 USB to UART Bridge Controller
[12076998.481275] usb 1-1.1: Manufacturer: Silicon Labs
[12076998.481278] usb 1-1.1: SerialNumber: IC-7300 02013763
[12076998.483782] cp210x 1-1.1:1.0: cp210x converter detected
[12076998.485872] usb 1-1.1: cp210x converter now attached to ttyUSB1
[12076998.549525] usb 1-1.4: new full-speed USB device number 27 using xhci_hcd
[12076998.628687] usb 1-1.4: New USB device found, idVendor=08bb, idProduct=2901, bcdDevice= 1.00
[12076998.628692] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[12076998.628695] usb 1-1.4: Product: USB Audio CODEC
[12076998.628698] usb 1-1.4: Manufacturer: Burr-Brown from TI
[12076998.650484] input: Burr-Brown from TI USB Audio CODEC as /devices/pci0000: … /input120
[12076998.703236] hid-generic 0003:08BB:2901.0042: input,hidraw3: USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC ]
                  on usb-0000:00:14.0-1.4/input3
```

ICARC Using the Raspberry PI

Presentation by KC0JFQ

Finding the CODEC and Serial:

dmesg: (ICOM7300 on PI3B+)

```
[ 1644.055818] usb 1-1.1.2: new full-speed USB device number 24 using dwc_otg
[ 1644.187708] usb 1-1.1.2: New USB device found, idVendor=0451, idProduct=2046
[ 1644.187724] usb 1-1.1.2: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[ 1644.188577] hub 1-1.1.2:1.0: USB hub found
[ 1644.189071] hub 1-1.1.2:1.0: 4 ports detected
[ 1644.515803] usb 1-1.1.2.1: new full-speed USB device number 25 using dwc_otg
[ 1644.664793] usb 1-1.1.2.1: New USB device found, idVendor=10c4, idProduct=ea60
[ 1644.664802] usb 1-1.1.2.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1644.664806] usb 1-1.1.2.1: Product: CP2102 USB to UART Bridge Controller
[ 1644.664810] usb 1-1.1.2.1: Manufacturer: Silicon Labs
[ 1644.664814] usb 1-1.1.2.1: SerialNumber: IC-7300 02013763
[ 1644.669329] cp210x 1-1.1.2.1:1.0: cp210x converter detected
[ 1644.672385] usb 1-1.1.2.1: cp210x converter now attached to ttyUSB0
[ 1644.765808] usb 1-1.1.2.4: new full-speed USB device number 26 using dwc_otg
[ 1644.918416] usb 1-1.1.2.4: New USB device found, idVendor=08bb, idProduct=2901
[ 1644.918424] usb 1-1.1.2.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1644.918428] usb 1-1.1.2.4: Product: USB Audio CODEC
[ 1644.918432] usb 1-1.1.2.4: Manufacturer: Burr-Brown from TI
[ 1644.938562] input: Burr-Brown from TI USB Audio CODEC  as /devices/platform/soc … /input7
[ 1645.006064] hid-generic 0003:08BB:2901.0008: input,hidraw3: USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC ]
               on usb-3f980000.usb-1.1.2.4/input3
```

dmesg: (KC0JFQ Interface on the PI3B+)

```
[ 2717.720758] usb 1-1.1.2: new full-speed USB device number 27 using dwc_otg
[ 2717.854074] usb 1-1.1.2: New USB device found, idVendor=0451, idProduct=2036
[ 2717.854090] usb 1-1.1.2: New USB device strings: Mfr=0, Product=1, SerialNumber=0
[ 2717.854099] usb 1-1.1.2: Product: General Purpose USB Hub
[ 2717.857399] hub 1-1.1.2:1.0: USB hub found
[ 2717.857956] hub 1-1.1.2:1.0: 3 ports detected
[ 2718.170720] usb 1-1.1.2.1: new full-speed USB device number 28 using dwc_otg
[ 2718.312666] usb 1-1.1.2.1: not running at top speed; connect to a high speed hub
[ 2718.323788] usb 1-1.1.2.1: New USB device found, idVendor=0403, idProduct=6011
[ 2718.323797] usb 1-1.1.2.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 2718.323801] usb 1-1.1.2.1: Product: KC0JFQ Radio I/F
[ 2718.323806] usb 1-1.1.2.1: Manufacturer: Perf Proc
[ 2718.323809] usb 1-1.1.2.1: SerialNumber: 20Z7D4IU
[ 2718.327333] ftdi_sio 1-1.1.2.1:1.0: FTDI USB Serial Device converter detected
[ 2718.327423] usb 1-1.1.2.1: Detected FT4232H
[ 2718.328178] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB0
[ 2718.334929] ftdi_sio 1-1.1.2.1:1.1: FTDI USB Serial Device converter detected
[ 2718.335031] usb 1-1.1.2.1: Detected FT4232H
[ 2718.335775] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB1
[ 2718.338857] ftdi_sio 1-1.1.2.1:1.2: FTDI USB Serial Device converter detected
[ 2718.338914] usb 1-1.1.2.1: Detected FT4232H
[ 2718.339608] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB2
[ 2718.342681] ftdi_sio 1-1.1.2.1:1.3: FTDI USB Serial Device converter detected
[ 2718.342769] usb 1-1.1.2.1: Detected FT4232H
[ 2718.343535] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB3
[ 2718.440721] usb 1-1.1.2.2: new full-speed USB device number 29 using dwc_otg
[ 2718.593174] usb 1-1.1.2.2: New USB device found, idVendor=08bb, idProduct=29b6
[ 2718.593184] usb 1-1.1.2.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 2718.593188] usb 1-1.1.2.2: Product: USB Audio CODEC
[ 2718.593191] usb 1-1.1.2.2: Manufacturer: Burr-Brown from TI
[ 2718.613239] input: Burr-Brown from TI USB Audio CODEC  as /devices/platform/  … input8
[ 2718.681076] hid-generic 0003:08BB:29B6.0009: input,hidraw3:
                USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC ] on usb-3f980000.usb-1.1.2.2/input3
```

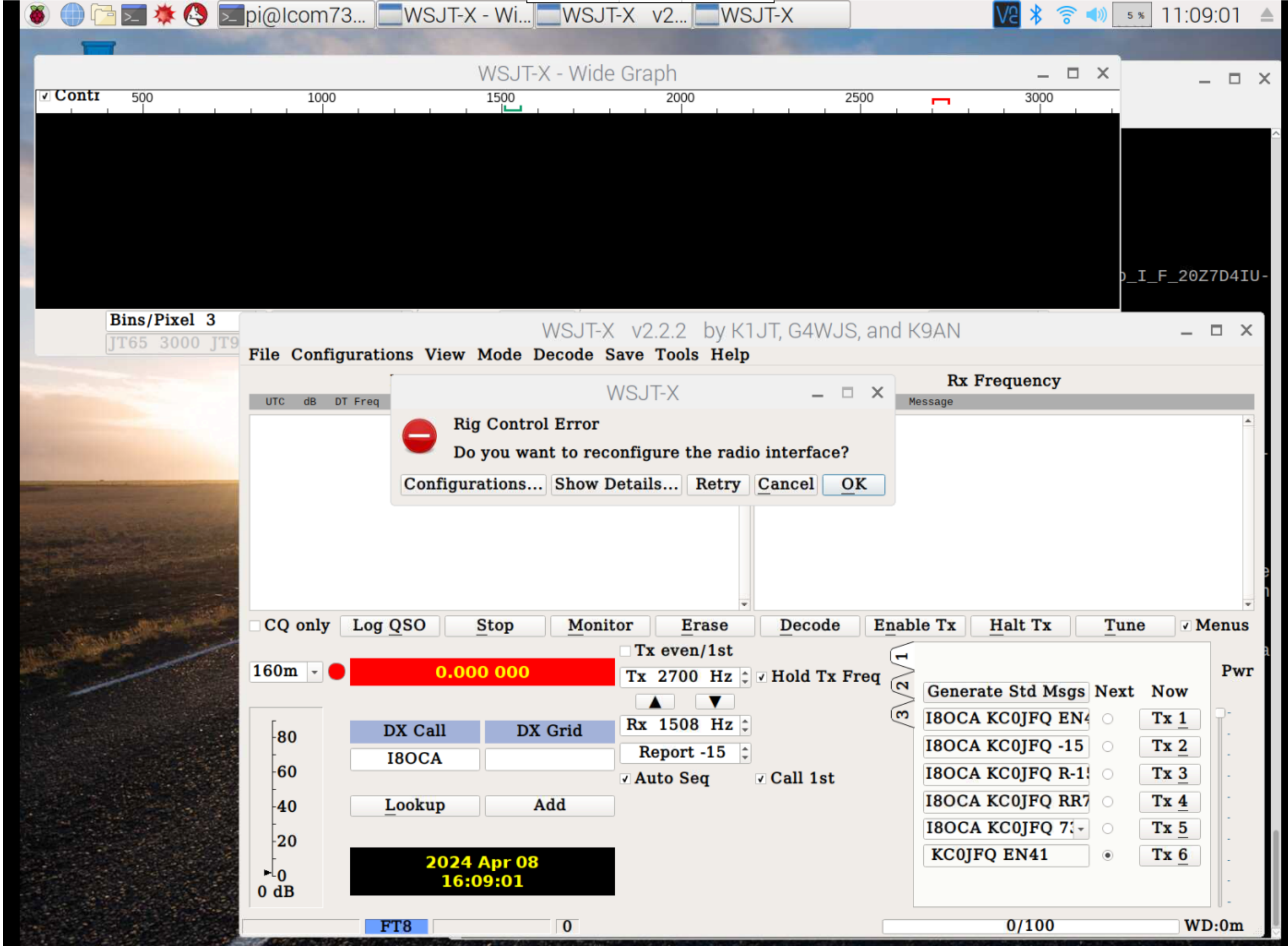
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 16

WSJTX

Operating without a radio connected



WSJT-X - Wide Graph

WSJT-X v2.2.2 by K1JT, G4WJS, and K9AN

WSJT-X

Rig Control Error

Do you want to reconfigure the radio interface?

Configurations... | Show Details... | Retry | Cancel | OK

File Configurations View Mode Decode Save Tools Help

□ CQ only | Log QSO | Stop | Monitor | Erase | Decode | Enable Tx | Halt Tx | Tune | ☑ Menus

Bins/Pixel 3

160m

0.000 000

□ Tx even/1st

Tx 2700 Hz

☑ Hold Tx Freq

Rx 1508 Hz

Report -15

☑ Auto Seq

☑ Call 1st

DX Call: I8OCA

DX Grid:

2024 Apr 08
16:09:01

Generate Std Msgs	Next	Now
I8OCA KC0JFQ EN4	○	Tx 1
I8OCA KC0JFQ -15	○	Tx 2
I8OCA KC0JFQ R-1!	○	Tx 3
I8OCA KC0JFQ RR7	○	Tx 4
I8OCA KC0JFQ 73	○	Tx 5
KC0JFQ EN41	●	Tx 6

FT8

0/100

WD:0m

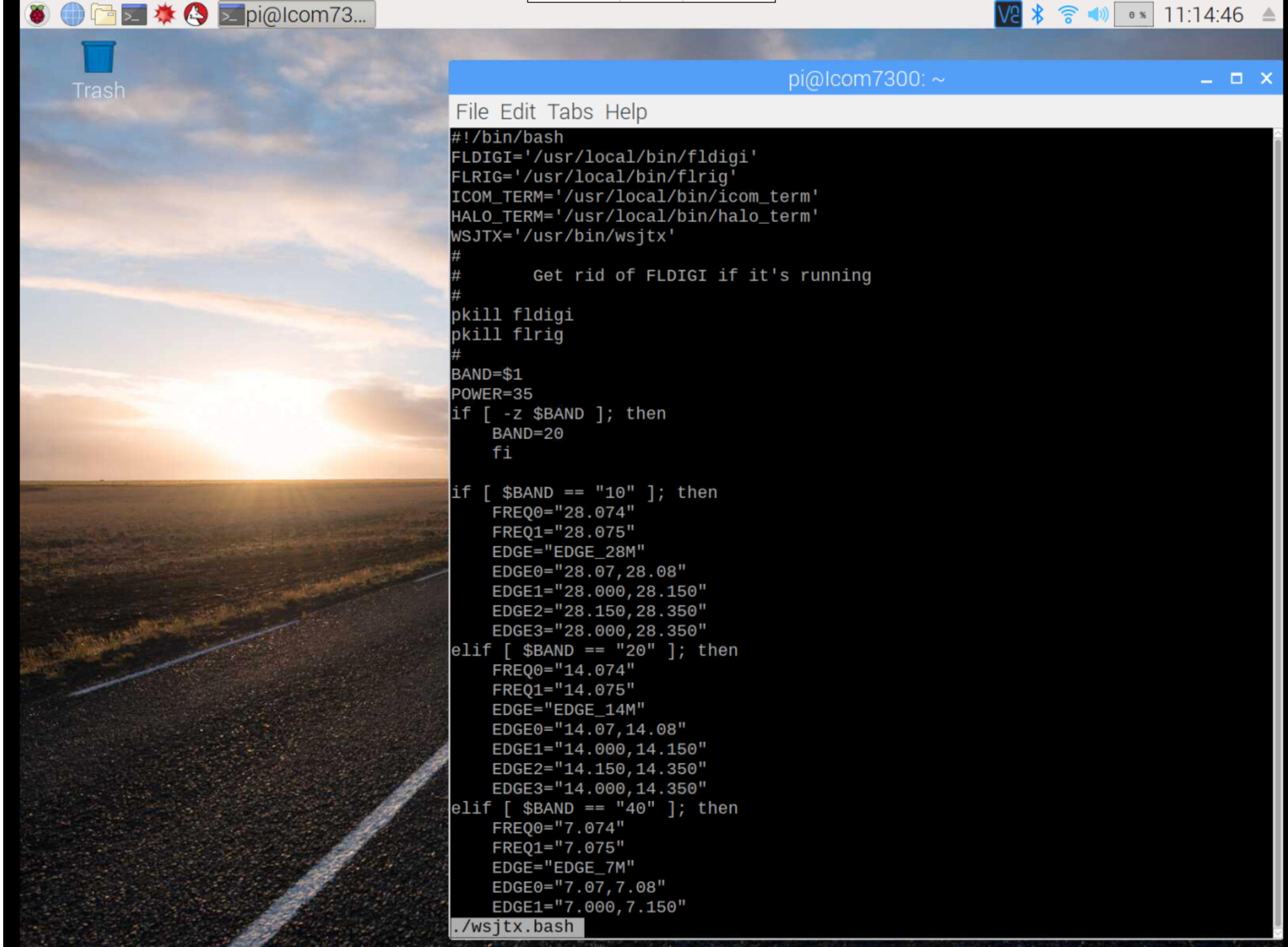
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 17

Starting WSJTX

Shell script



```
#!/bin/bash
FLDIGI='/usr/local/bin/fldigi'
FLRIG='/usr/local/bin/flrig'
ICOM_TERM='/usr/local/bin/icom_term'
HALO_TERM='/usr/local/bin/halo_term'
WSJTX='/usr/bin/wsjtx'
#
#       Get rid of FLDIGI if it's running
#
pkill fldigi
pkill flrig
#
BAND=$1
POWER=35
if [ -z $BAND ]; then
    BAND=20
    fi

if [ $BAND == "10" ]; then
    FREQ0="28.074"
    FREQ1="28.075"
    EDGE="EDGE_28M"
    EDGE0="28.07,28.08"
    EDGE1="28.000,28.150"
    EDGE2="28.150,28.350"
    EDGE3="28.000,28.350"
elif [ $BAND == "20" ]; then
    FREQ0="14.074"
    FREQ1="14.075"
    EDGE="EDGE_14M"
    EDGE0="14.07,14.08"
    EDGE1="14.000,14.150"
    EDGE2="14.150,14.350"
    EDGE3="14.000,14.350"
elif [ $BAND == "40" ]; then
    FREQ0="7.074"
    FREQ1="7.075"
    EDGE="EDGE_7M"
    EDGE0="7.07,7.08"
    EDGE1="7.000,7.150"
./wsjtx.bash
```

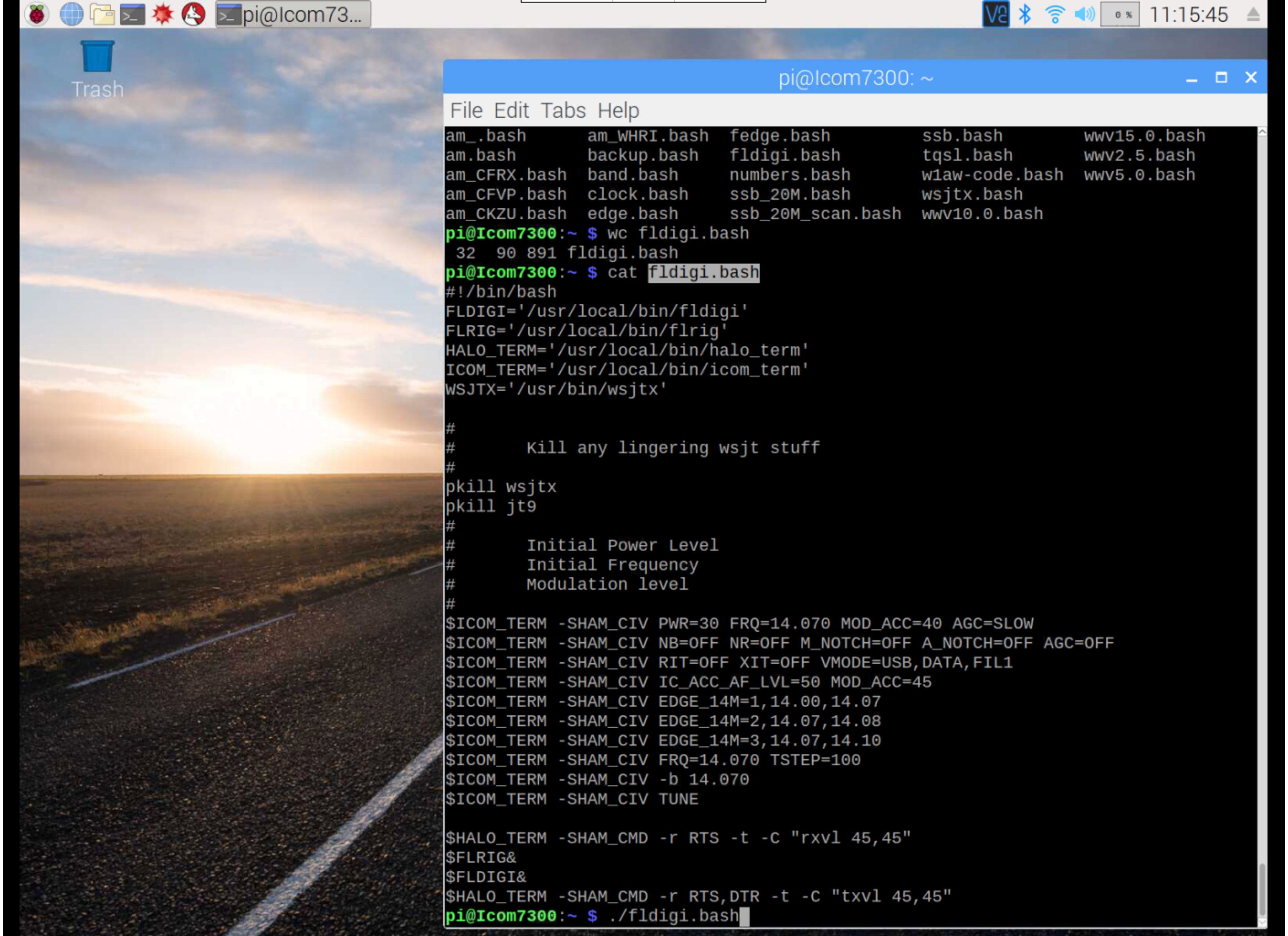
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 18

Starting FLDIGI (PSK31)

Shell script



```
am_.bash        am_WHRI.bash   fedge.bash          ssb.bash        wwv15.0.bash
am.bash         backup.bash    fldigi.bash         tqsl.bash       wwv2.5.bash
am_CFRX.bash    band.bash      numbers.bash        w1aw-code.bash  wwv5.0.bash
am_CFVP.bash    clock.bash     ssb_20M.bash        wsjtx.bash
am_CKZU.bash    edge.bash      ssb_20M_scan.bash   wwv10.0.bash
pi@Icom7300:~ $ wc fldigi.bash
 32  90 891 fldigi.bash
pi@Icom7300:~ $ cat fldigi.bash
#!/bin/bash
FLDIGI='/usr/local/bin/fldigi'
FLRIG='/usr/local/bin/flrig'
HALO_TERM='/usr/local/bin/halo_term'
ICOM_TERM='/usr/local/bin/icom_term'
WSJTX='/usr/bin/wsjtx'

#
#       Kill any lingering wsjt stuff
#
pkill wsjtx
pkill jt9
#
#       Initial Power Level
#       Initial Frequency
#       Modulation level
#
$ICOM_TERM -SHAM_CIV PWR=30 FRQ=14.070 MOD_ACC=40 AGC=SLOW
$ICOM_TERM -SHAM_CIV NB=OFF NR=OFF M_NOTCH=OFF A_NOTCH=OFF AGC=OFF
$ICOM_TERM -SHAM_CIV RIT=OFF XIT=OFF VMODE=USB,DATA,FIL1
$ICOM_TERM -SHAM_CIV IC_ACC_AF_LVL=50 MOD_ACC=45
$ICOM_TERM -SHAM_CIV EDGE_14M=1,14.00,14.07
$ICOM_TERM -SHAM_CIV EDGE_14M=2,14.07,14.08
$ICOM_TERM -SHAM_CIV EDGE_14M=3,14.07,14.10
$ICOM_TERM -SHAM_CIV FRQ=14.070 TSTEP=100
$ICOM_TERM -SHAM_CIV -b 14.070
$ICOM_TERM -SHAM_CIV TUNE

$HALO_TERM -SHAM_CMD -r RTS -t -C "rxvl 45,45"
$FLRIG&
$FLDIGI&
$HALO_TERM -SHAM_CMD -r RTS,DTR -t -C "txvl 45,45"
pi@Icom7300:~ $ ./fldigi.bash
```

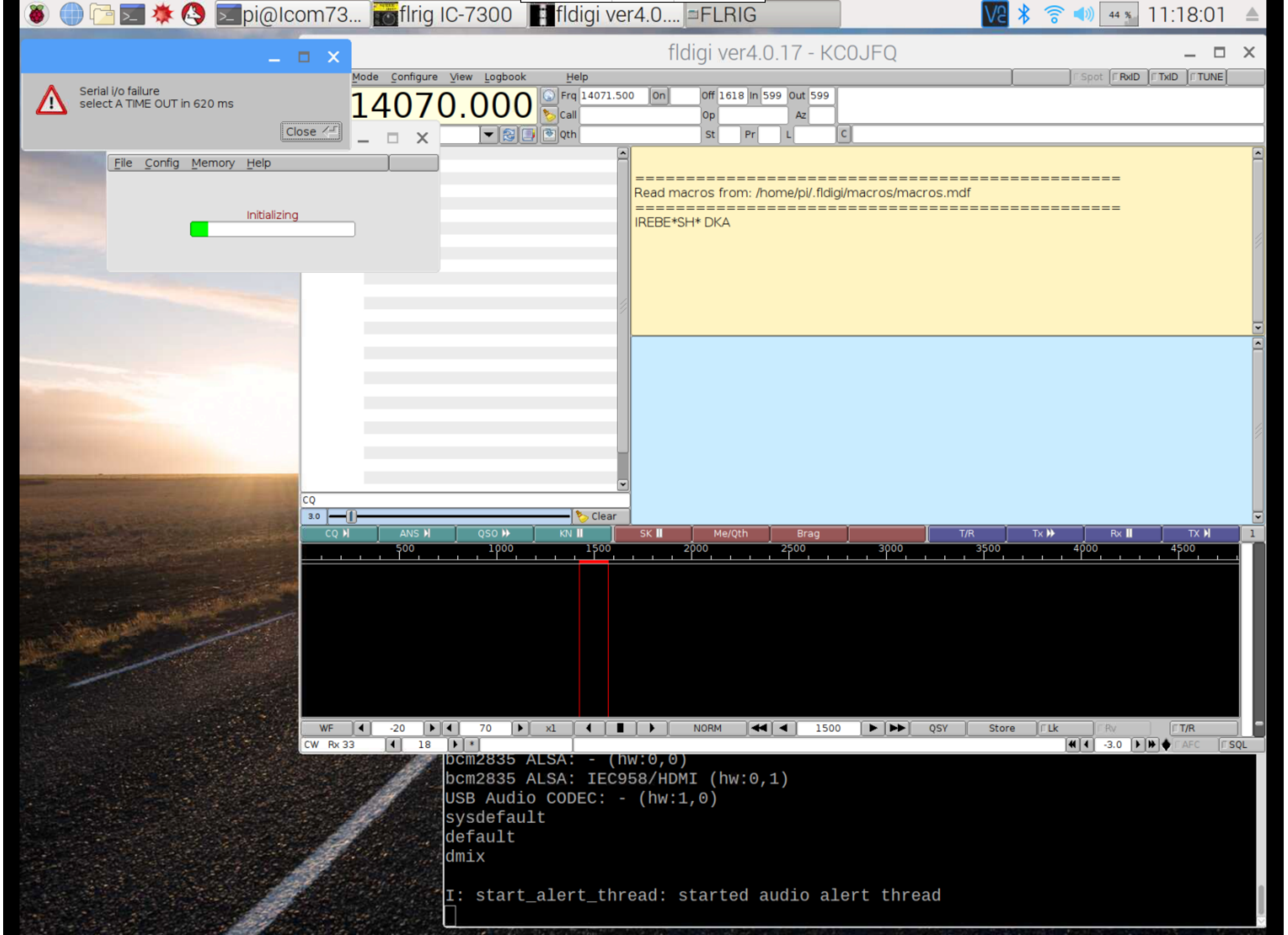
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 19

Running FLDIGI

No radio error message



Serial i/o failure
select A TIME OUT in 620 ms

Close

Initializing

fldigi ver4.0.17 - KC0JFQ

14070.000

Frq 14071.500 On Off 1618 In 599 Out 599
Call Op Az
Qth St Pr L C

```
====================================================
Read macros from: /home/pi/.fldigi/macros/macros.mdf
====================================================
IREBE*SH* DKA
```

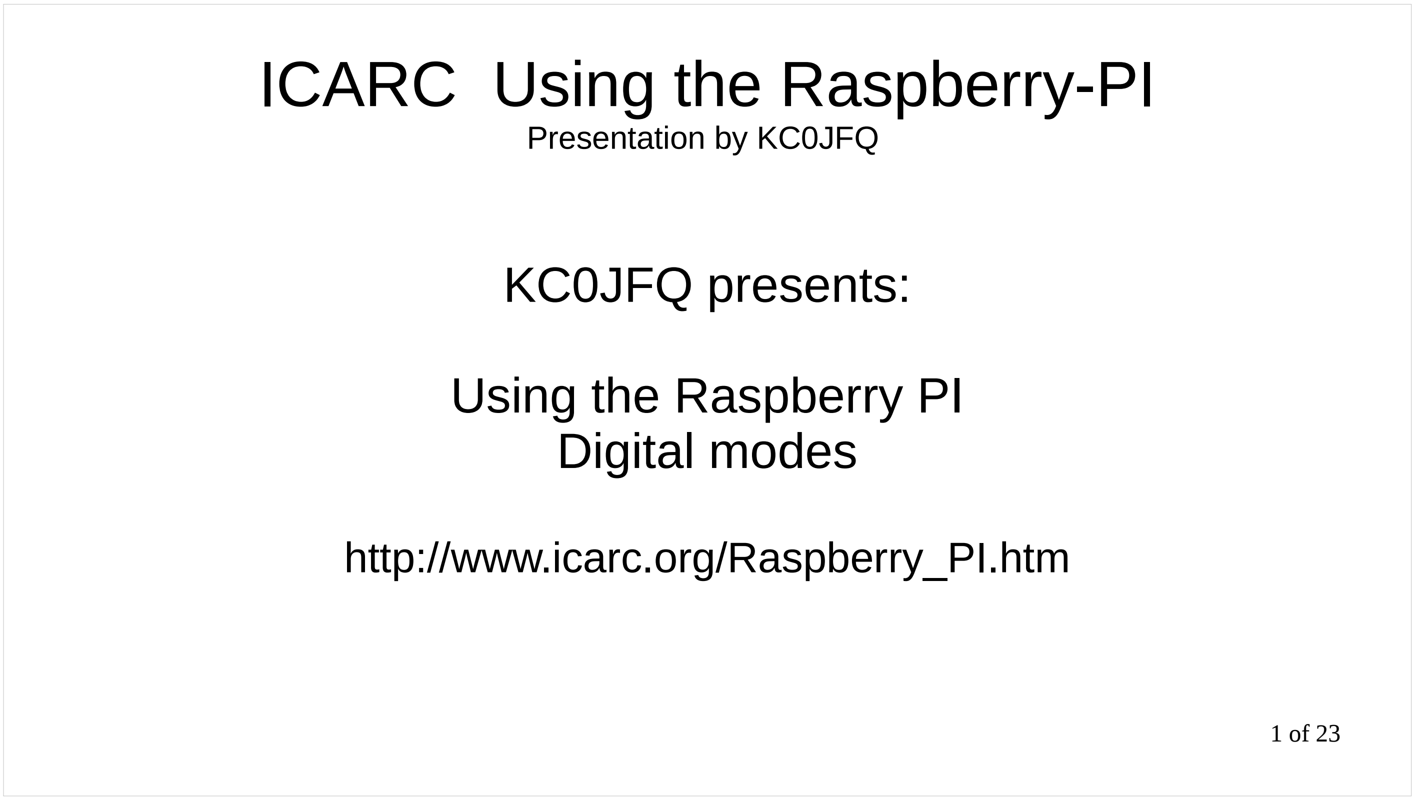
```
bcm2835 ALSA: - (hw:0,0)
bcm2835 ALSA: IEC958/HDMI (hw:0,1)
USB Audio CODEC: - (hw:1,0)
sysdefault
default
dmix

I: start_alert_thread: started audio alert thread
```

ICARC

Using the Raspberry PI

Presentation by KC0JFQ

Out of things to put on slides



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents:

Using the Raspberry PI
Digital modes

http://www.icarc.org/Raspberry_PI.htm

Here we go again!

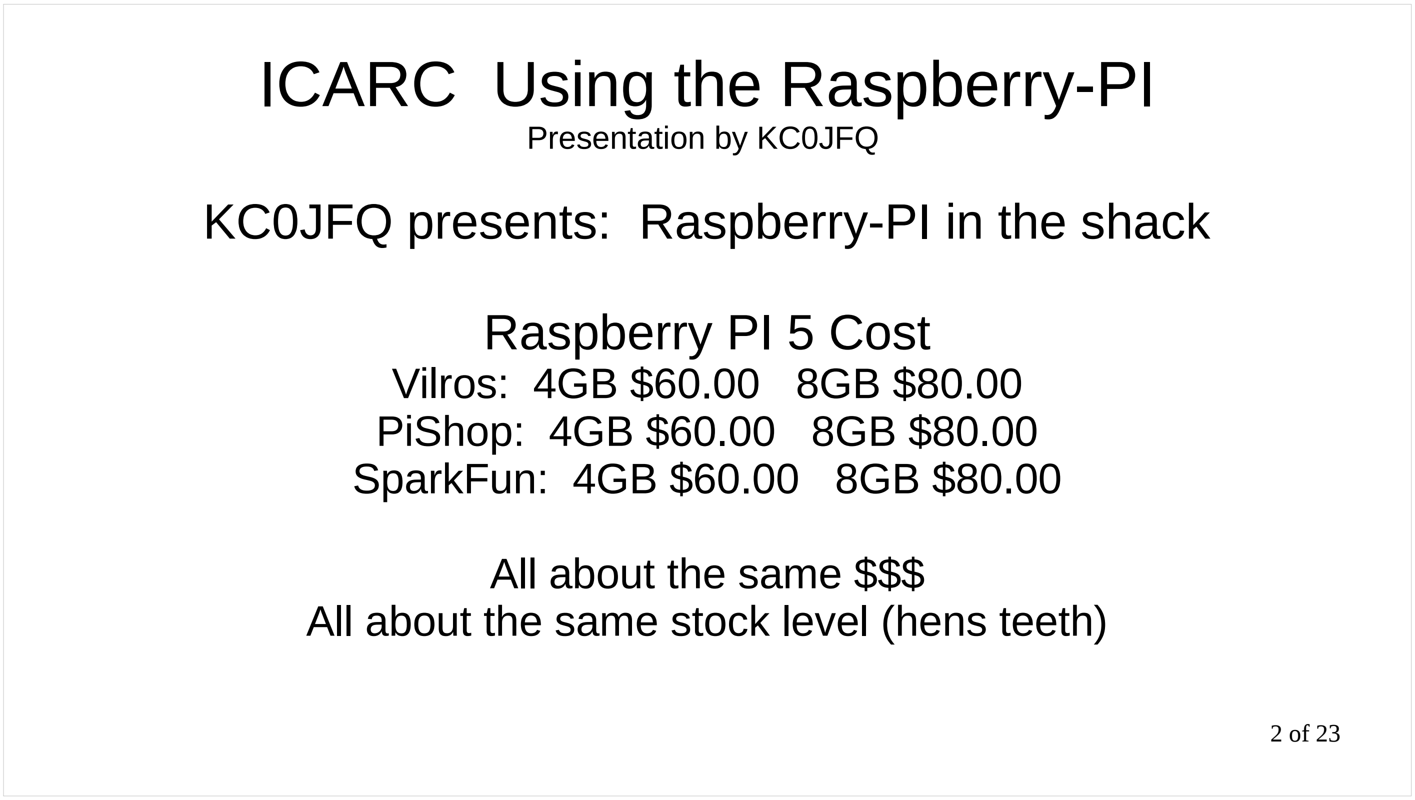
Discussion of the Raspberry-PI.

My experience using the Raspberry-PI for digital modes.

I rarely upgrade the PI.

I want to turn it on and use it.

Software upgrade can break things and take everything down.



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI 5 Cost

Vilros: 4GB $60.00 8GB $80.00
PiShop: 4GB $60.00 8GB $80.00
SparkFun: 4GB $60.00 8GB $80.00

All about the same $$$
All about the same stock level (hens teeth)

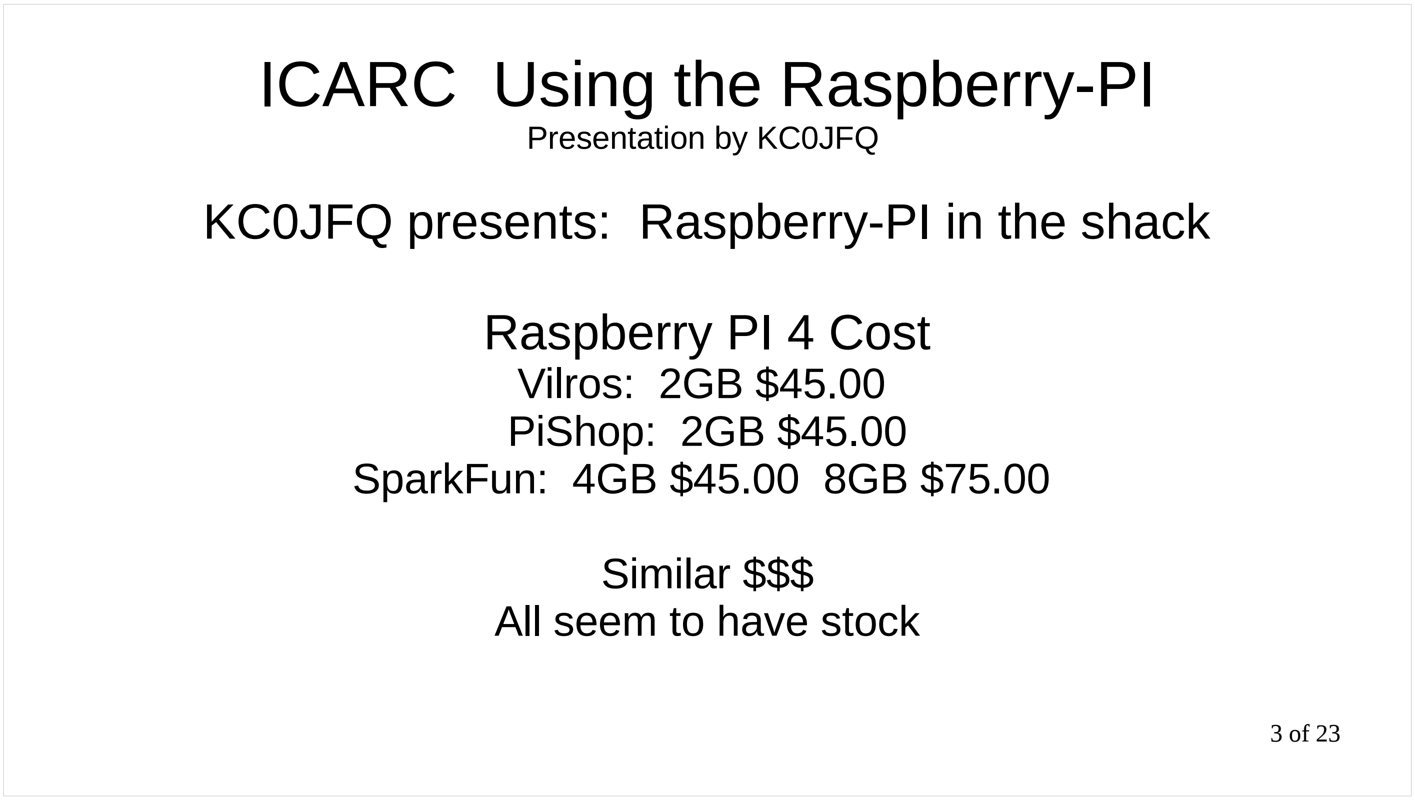
They used to be a flat $40.00, but that was 10 years ago.

Not too much more expensive but lots more memory, CPU speed, and USB speed with PI3/PI4.

PI5 is another leap in performance, but probably won’t make much difference for digital modes.

Plan on using USB interface in your radio. This offloads the PI CPU.

Good Example is KC0JFQ audio interface.



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI 4 Cost

Vilros: 2GB $45.00
PiShop: 2GB $45.00
SparkFun: 4GB $45.00 8GB $75.00

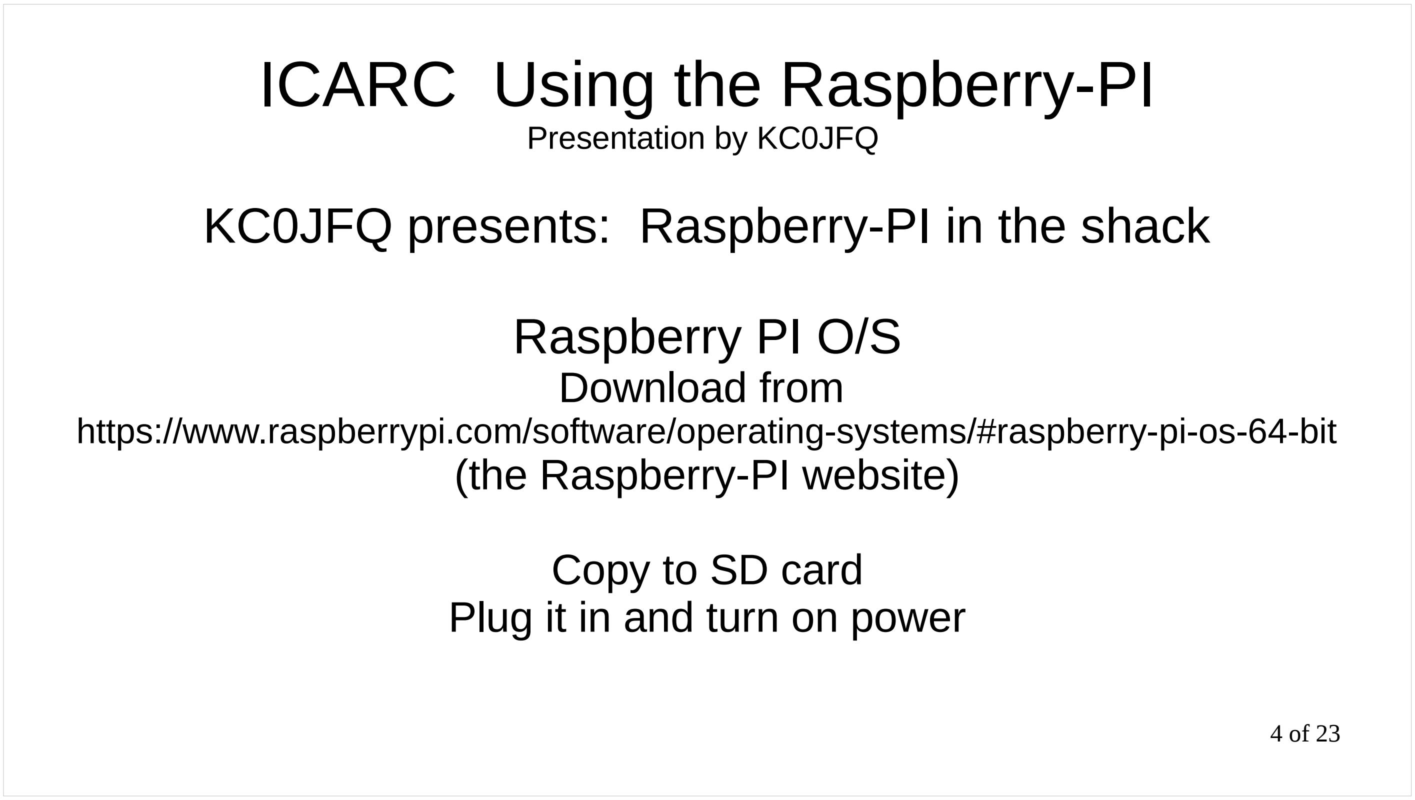
Similar $$$
All seem to have stock

PI4 and PI3 are more readily available.

I don’t think a PI-ZERO would work too well. Awfully sparse I/O *stuff*. Speed probably marginal as well.

The Raspberry-PI zero was used in three of our fox transmitters. For that application the PI is a power pig. BUT it rans for hours on 6 AAA cells!

My setup is a 2017 vintage Raspberry-PI3 B+



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI O/S

Download from
https://www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-64-bit
(the Raspberry-PI website)

Copy to SD card
Plug it in and turn on power

Several ways to install O/S.

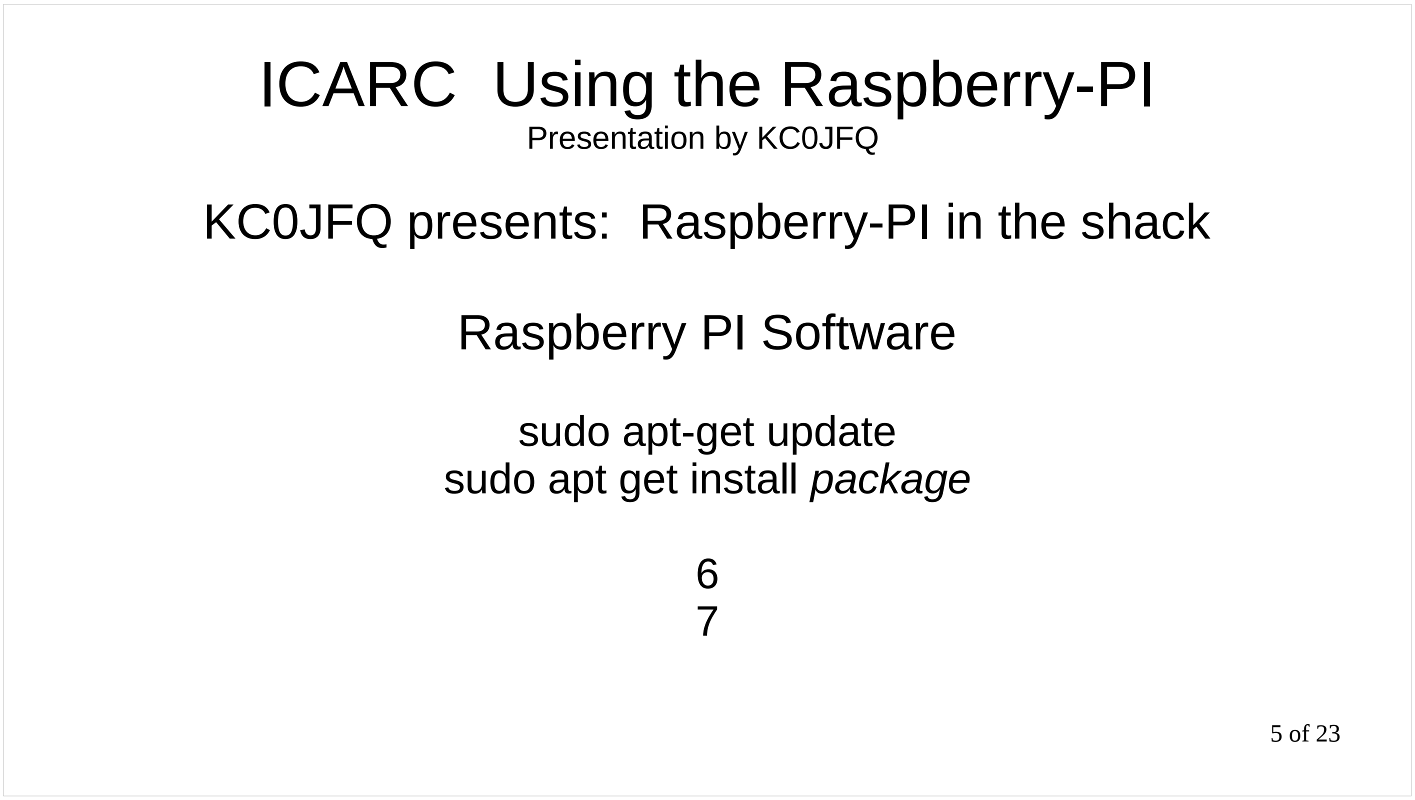
I download SD image to big machine and copy it to SD card there.

Additional methods available now, but why change.

I use linux, so the SD imaging commands used for 20+ years work the same.

Boot with the downloaded image and expand filesystem. The downloaded image is compressed and crammed onto an 8MB or 16MB SDCard. Get a big one, 64GB or larger that is fast. This is your system disk, so read speed is critical.

Backup expanded SD card onto the big machine to save your work (maybe onto a blue-ray disc???)



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Software

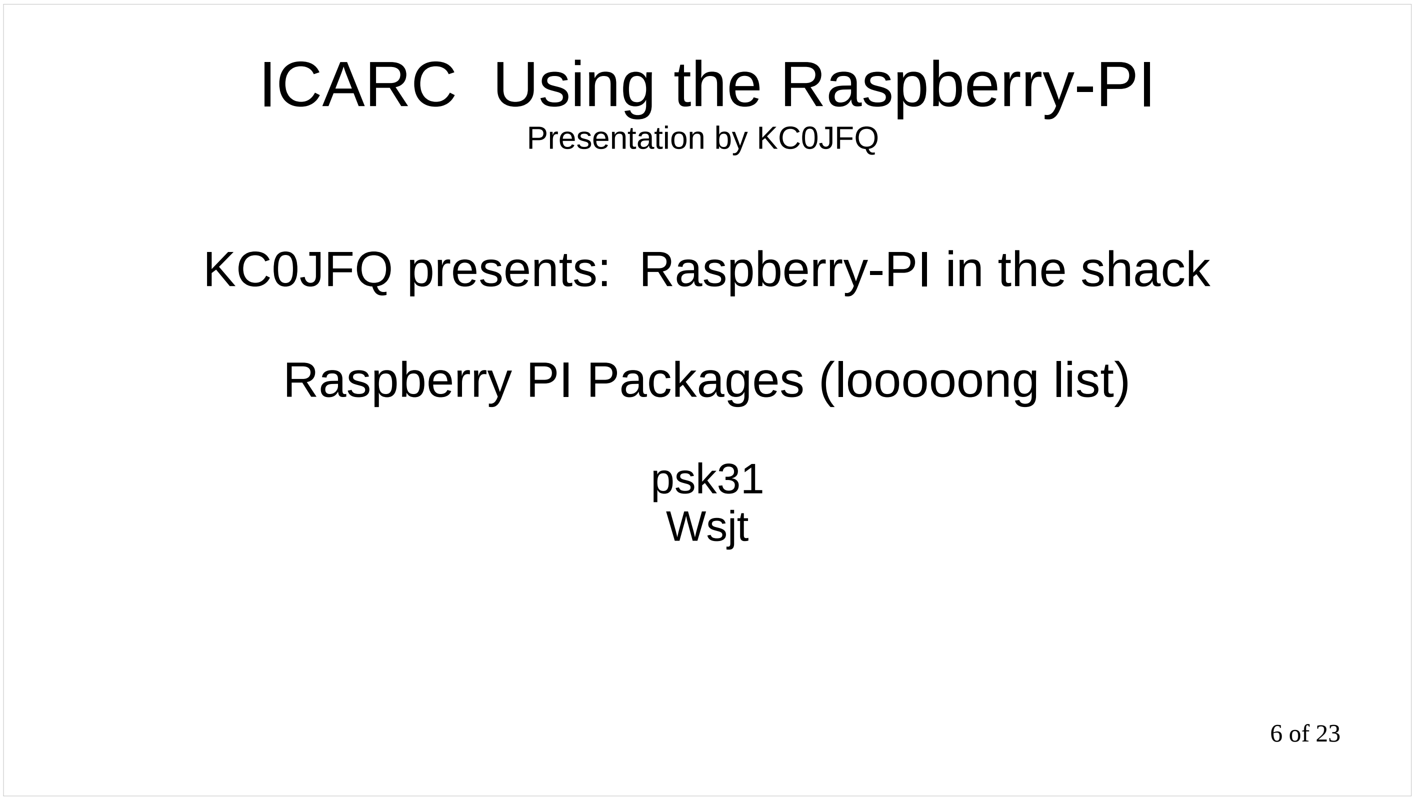
```
sudo apt-get update
sudo apt get install package
```

6
7

Once your O/S is loaded and you’ve configured connectivity, use the install application to load all yhe packages you’ll need.

Many ham packages are pre-packaged for the Raspberry-PI and will install this way.

Some of them are source distributions that use **make** or **cmake** to build and install. Once you get pre-requisites loaded they build albeit a bit slowly compared to the big machine.



ICARC Using the Raspberry-PI

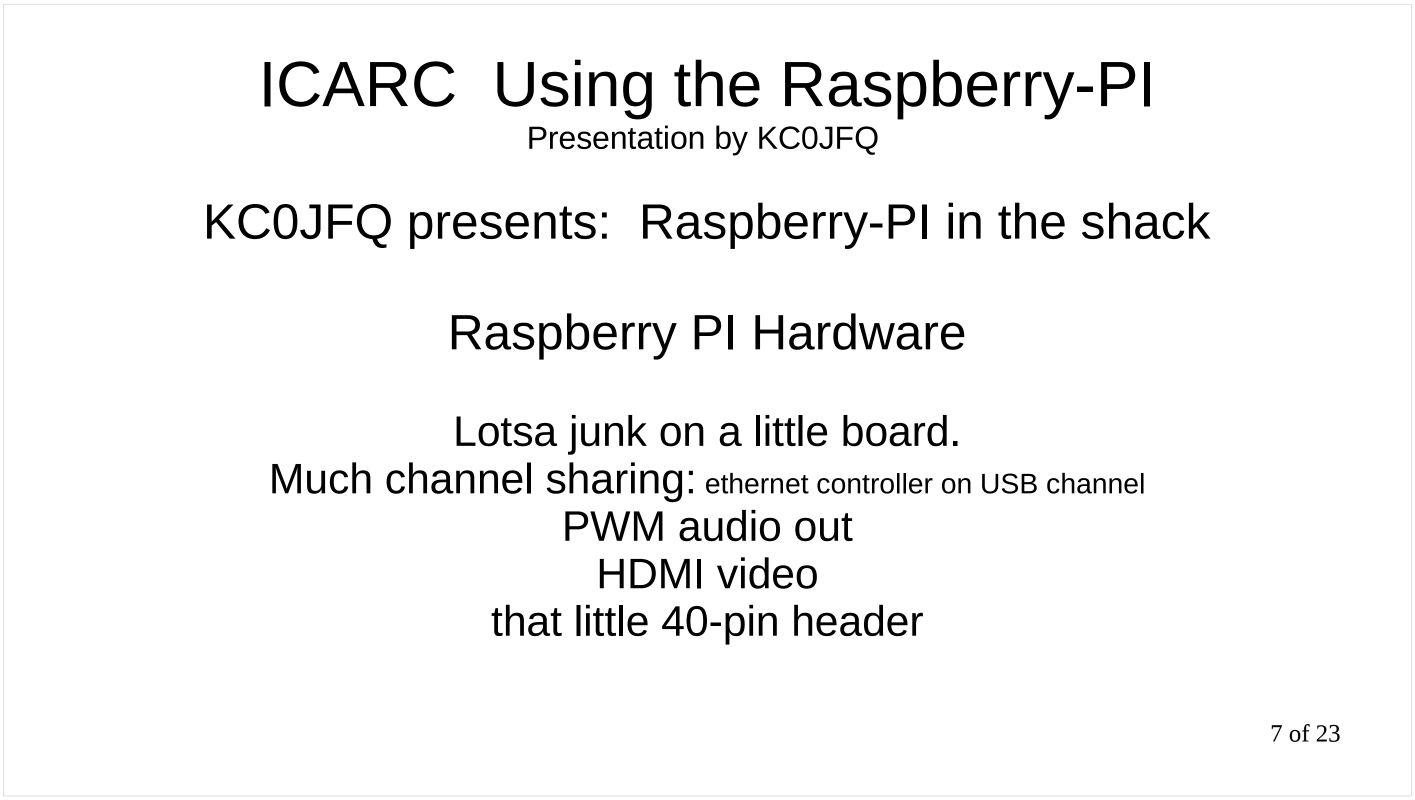
Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Packages (looooong list)

psk31
Wsjt

Look in notes to find out if these are packaged!



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Hardware

Lotsa junk on a little board.
Much channel sharing: ethernet controller on USB channel
PWM audio out
HDMI video
that little 40-pin header

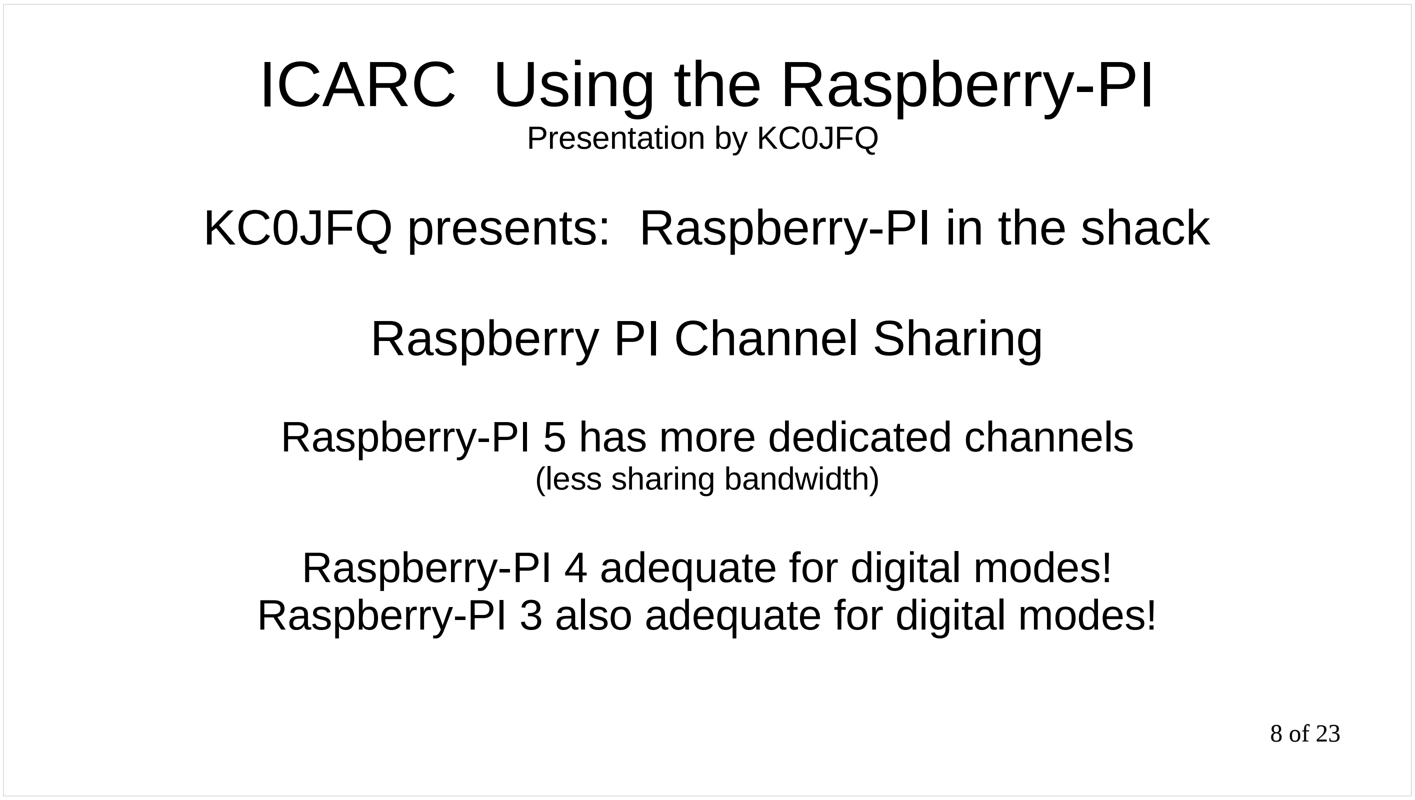
PWM audio works well, but it in one direction (out).

Also has IIS interface on 40-pin header. I think someone made a board with IIS audio chip(s). But just get a standard USB audio device (ancient Burr-Brown device PCM2906 from TI these days).

Most newer radios have this in-built with a CP2102 for serial control. We do both of these effortlessly.

HDMI port is micro-HDMI, so adapter cable or a simple adapter is required. USB keyboard works. Need video/keyboard for initial setup.

Use VNCVIEWER to connect you big computer to the PI. BUT VNCVIEWER is a PITA ‘cause authors are fighting over it :-(



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Channel Sharing

Raspberry-PI 5 has more dedicated channels
(less sharing bandwidth)

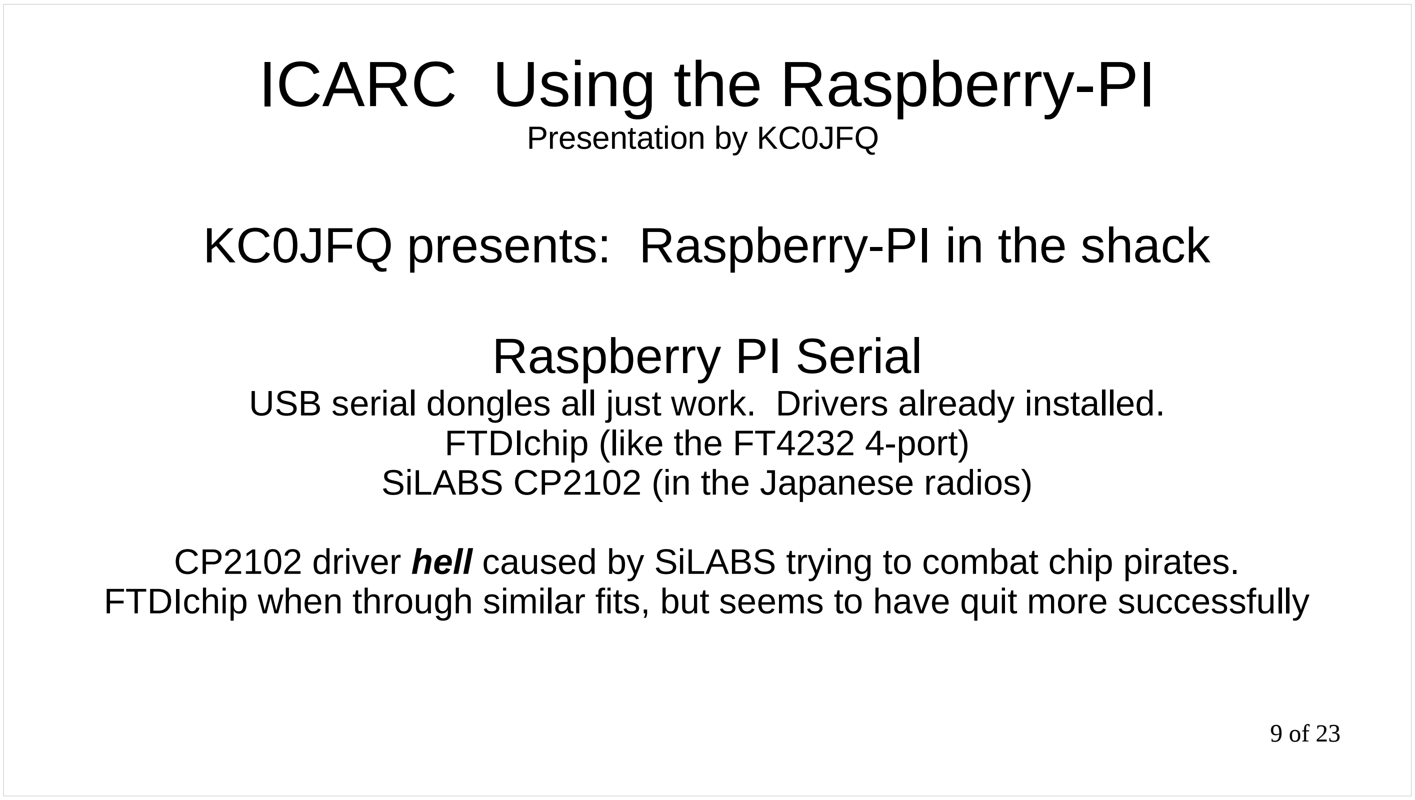
Raspberry-PI 4 adequate for digital modes!
Raspberry-PI 3 also adequate for digital modes!

8 of 23

Once the CPU cycles are available to do the required processing, more performance doesn’t buy us much.

Point of PI is it’s cheap and can be dedicated to run only your radio software. If you want to upgrade something, image the SD card or replace it and you can go back to a working setup very quickly.

If I’ve been lucky, I’ll fire up my PI3B+ on Monday before the meeting and remember how to control the IC7300.



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

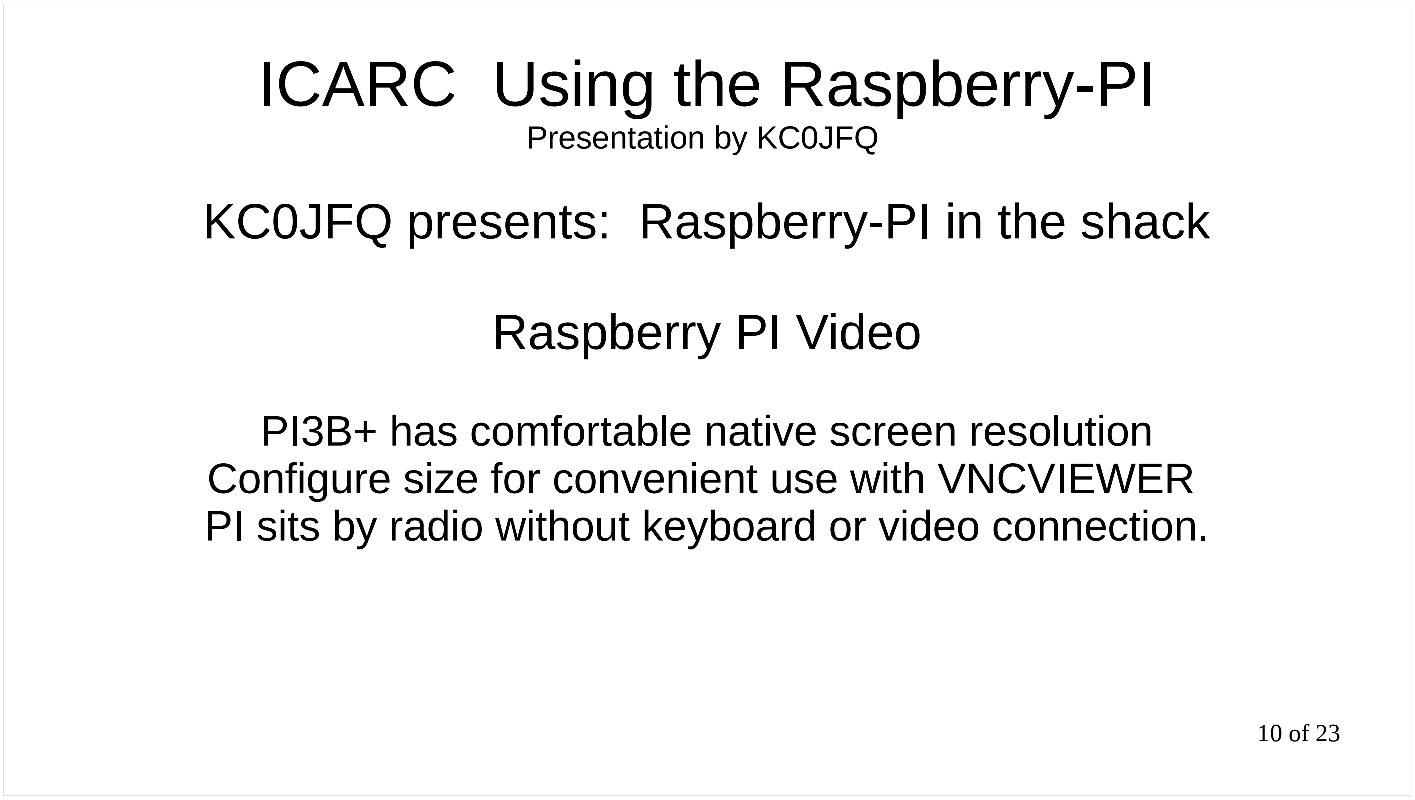
KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Serial

USB serial dongles all just work. Drivers already installed.
FTDIchip (like the FT4232 4-port)
SiLABS CP2102 (in the Japanese radios)

CP2102 driver ***hell*** caused by SiLABS trying to combat chip pirates.
FTDIchip when through similar fits, but seems to have quit more successfully

I’ve had problems with the CP2102 in the IC7300 on Windows systems. Bastard driver. Bastard install system. Windows is PITA with that chip. FTDIchip just works everywhere.



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

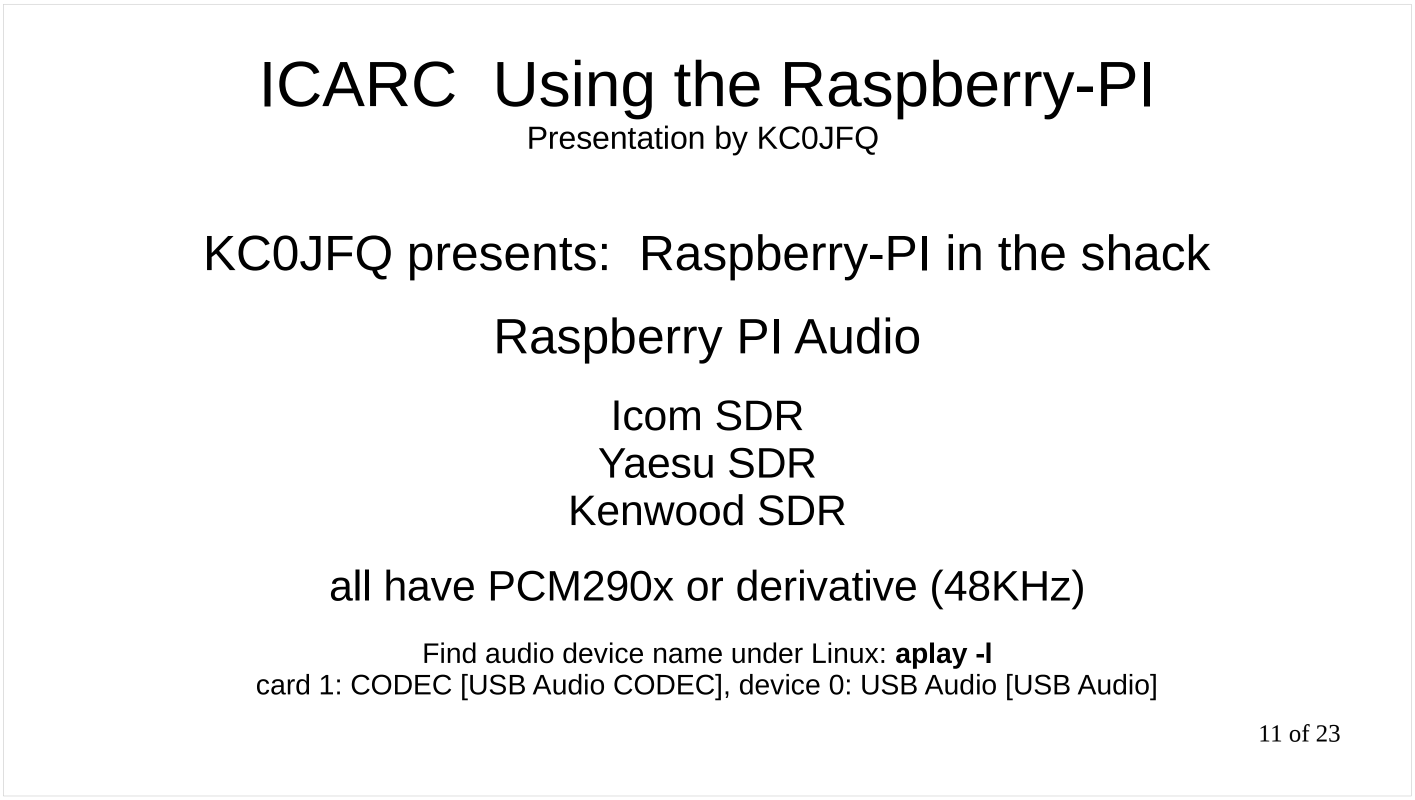
Raspberry PI Video

PI3B+ has comfortable native screen resolution
Configure size for convenient use with VNCVIEWER
PI sits by radio without keyboard or video connection.

10 of 23

A dedicated screen seems necessary to set up the Raspberry-PI (and a keyboard).

Switch over to VNCVIEWER once installed to eliminate clutter on the desk. Will have keyboard, mouse, and screen connection for presentation, but not used once PI has been configured.



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Audio

Icom SDR
Yaesu SDR
Kenwood SDR

all have PCM290x or derivative (48KHz)

Find audio device name under Linux: **aplay -l**

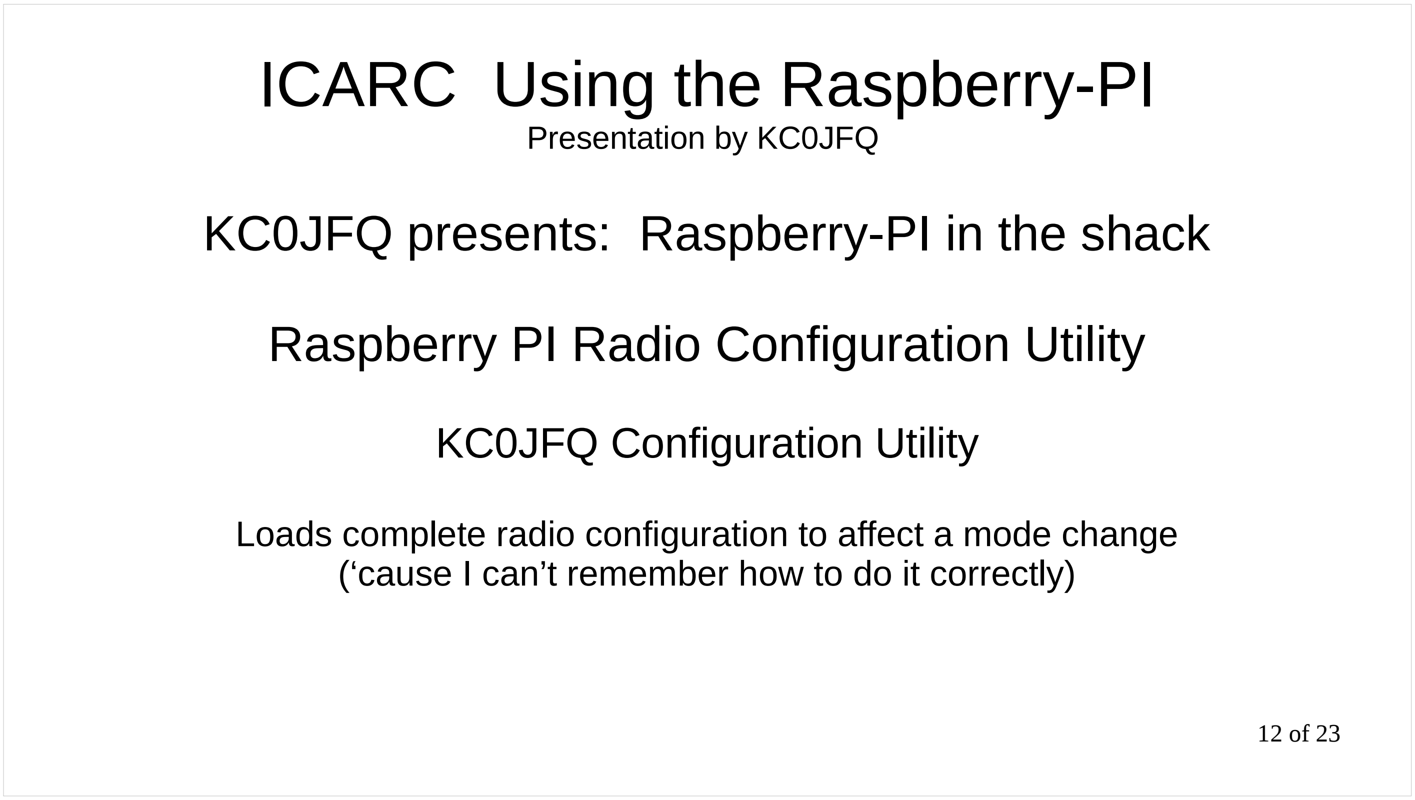
```
card 1: CODEC [USB Audio CODEC], device 0: USB Audio [USB Audio]
```

All current radios have what amounts to a sound-card and serial port built in.

You’ll probably find a PCM2906 (mentioned earlier) or similar. This is a 48KHz 16 bit 2-channel A/D and D/A. Pay attention to left-channel / right-channel settings. Easy configuration error selecting the wrong one. Probably set digital mode software to send both R and L channels.

They all seem to have that brain-dead CP2102 device as well. DRIVER HELL

BUT they all have a digital-mode connector (line level in and out).



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

KC0JFQ presents: Raspberry-PI in the shack

Raspberry PI Radio Configuration Utility

KC0JFQ Configuration Utility

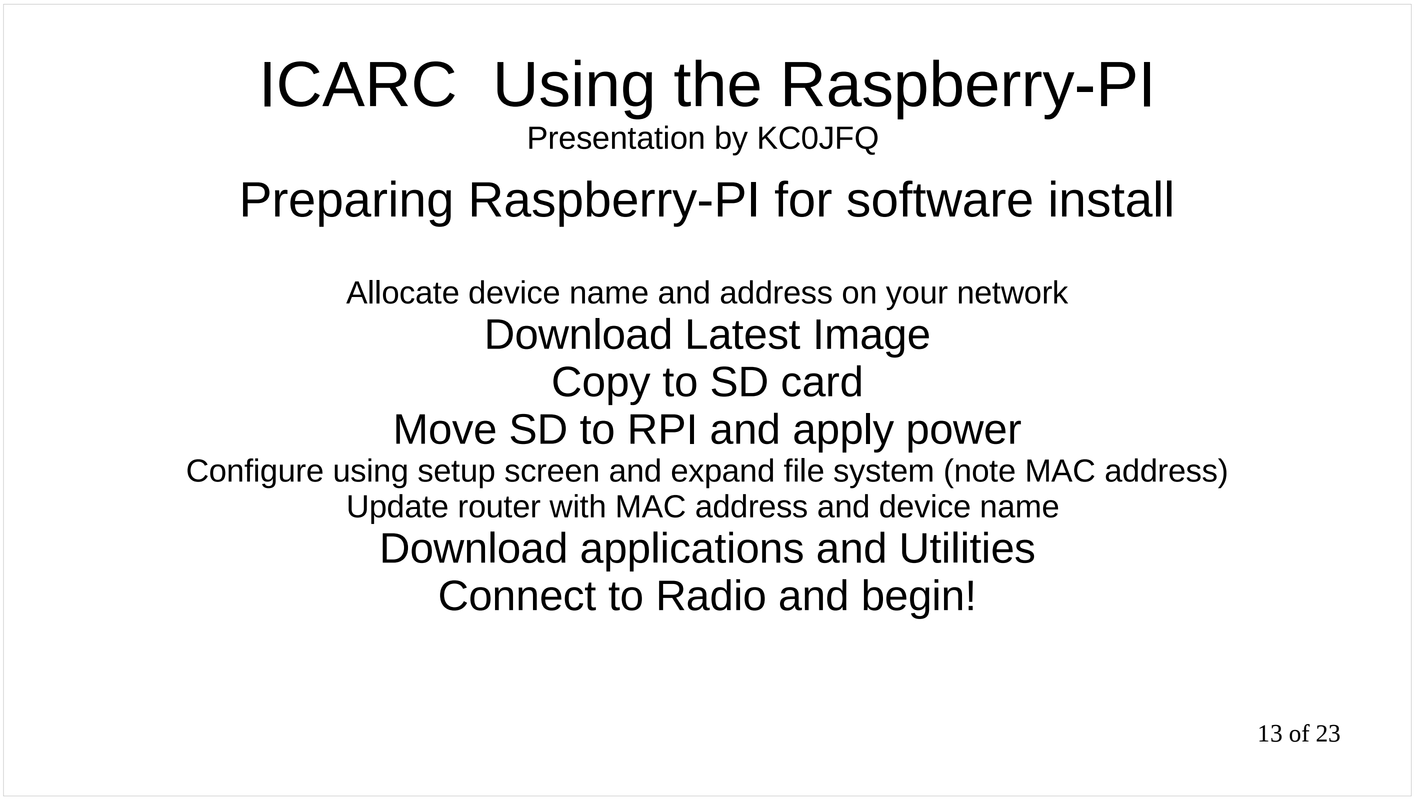
Loads complete radio configuration to affect a mode change
(‘cause I can’t remember how to do it correctly)

My setup is an ICOM7300 and the audio interface I designed and built back in 2013. IC7300 works equally well with a direct connect.

Shell script sends commands to my interface to mess with audio gain settings. The sends commands to radio to configure it for the digital mode at hand. The WSJTX script has 18 individual setup commands to the IC7300 ans 5 setup commands to the radio interface.

This way I don’t miss a critical setting.

The



ICARC Using the Raspberry-PI

Presentation by KC0JFQ

Preparing Raspberry-PI for software install

Allocate device name and address on your network
Download Latest Image
Copy to SD card
Move SD to RPI and apply power
Configure using setup screen and expand file system (note MAC address)
Update router with MAC address and device name
Download applications and Utilities
Connect to Radio and begin!

Allocate a fixed address for your PI if you plan on accessing it from other machines, otherwise it will wander around making things like VNCVIEWER a PITA.

Once you have the router updated power down the PI and then reboot the router. Once router is up you can power-up the PI. This power interruption forces everyone to use updated address data. Now when you power-on the PI, it will appear at a static address.

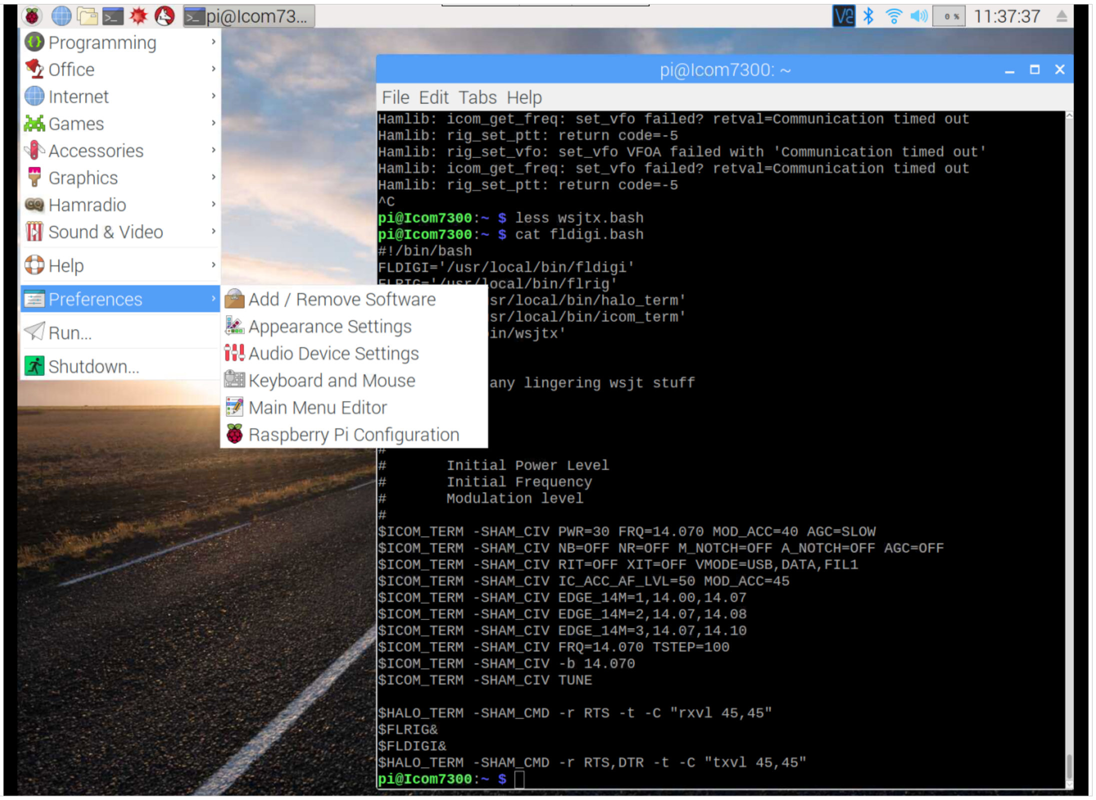
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 14

Raspberry-PI Menu

Chromium Web browser slow on PI3B+



```
Hamlib: icom_get_freq: set_vfo failed? retval=Communication timed out
Hamlib: rig_set_ptt: return code=-5
Hamlib: rig_set_vfo: set_vfo VFOA failed with 'Communication timed out'
Hamlib: icom_get_freq: set_vfo failed? retval=Communication timed out
Hamlib: rig_set_ptt: return code=-5
^C
pi@Icom7300:~ $ less wsjtx.bash
pi@Icom7300:~ $ cat fldigi.bash
#!/bin/bash
FLDIGI='/usr/local/bin/fldigi'
FLRIG='/usr/local/bin/flrig'
             sr/local/bin/halo_term'
             sr/local/bin/icom_term'
             in/wsjtx'


             any lingering wsjt stuff




#
#       Initial Power Level
#       Initial Frequency
#       Modulation level
#
$ICOM_TERM -SHAM_CIV PWR=30 FRQ=14.070 MOD_ACC=40 AGC=SLOW
$ICOM_TERM -SHAM_CIV NB=OFF NR=OFF M_NOTCH=OFF A_NOTCH=OFF AGC=OFF
$ICOM_TERM -SHAM_CIV RIT=OFF XIT=OFF VMODE=USB,DATA,FIL1
$ICOM_TERM -SHAM_CIV IC_ACC_AF_LVL=50 MOD_ACC=45
$ICOM_TERM -SHAM_CIV EDGE_14M=1,14.00,14.07
$ICOM_TERM -SHAM_CIV EDGE_14M=2,14.07,14.08
$ICOM_TERM -SHAM_CIV EDGE_14M=3,14.07,14.10
$ICOM_TERM -SHAM_CIV FRQ=14.070 TSTEP=100
$ICOM_TERM -SHAM_CIV -b 14.070
$ICOM_TERM -SHAM_CIV TUNE

$HALO_TERM -SHAM_CMD -r RTS -t -C "rxvl 45,45"
$FLRIG&
$FLDIGI&
$HALO_TERM -SHAM_CMD -r RTS,DTR -t -C "txvl 45,45"
pi@Icom7300:~ $ 
```

Here is the PI menu chased down a bit.

ICARC

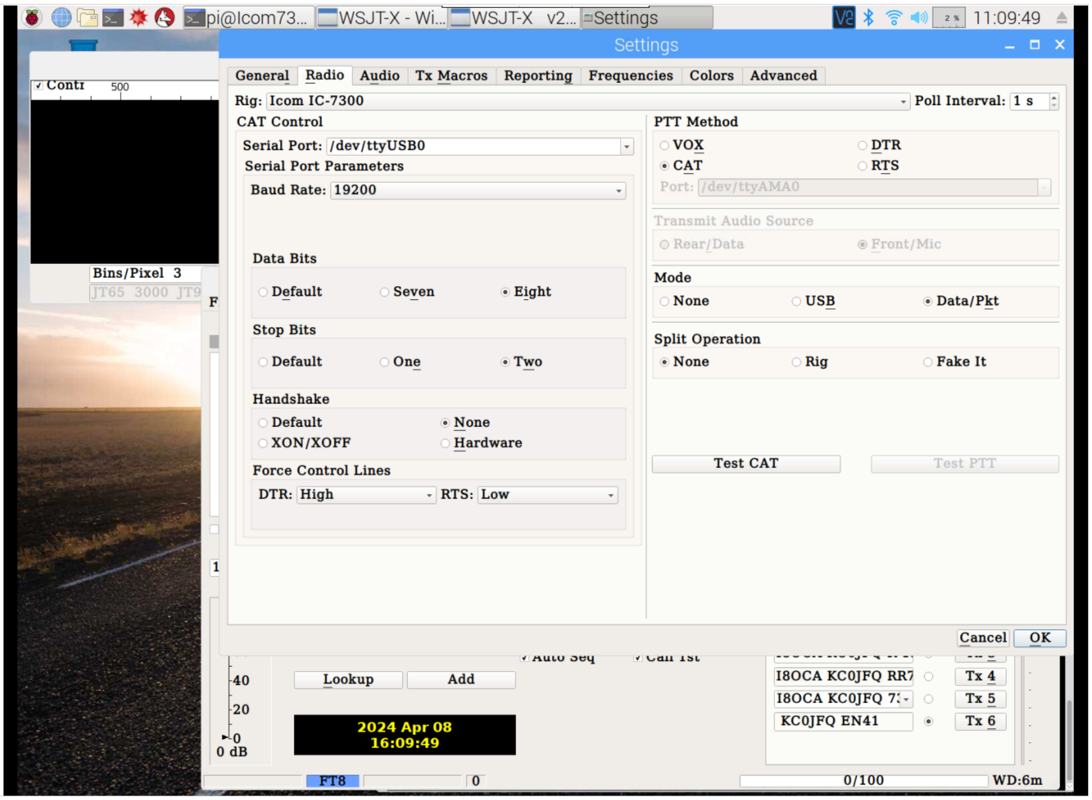
Using the Raspberry PI

Presentation by KC0JFQ
15

WSJTX

the setup screen

CARD=CODEC,DEV=0



Rig: Icom IC-7300 Poll Interval: 1 s

CAT Control

Serial Port: /dev/ttyUSB0

Serial Port Parameters

Baud Rate: 19200

Data Bits: ○ Default ○ Seven ● Eight

Stop Bits: ○ Default ○ One ● Two

Handshake: ○ Default ● None ○ XON/XOFF ○ Hardware

Force Control Lines: DTR: High RTS: Low

PTT Method: ○ VOX ○ DTR ● CAT ○ RTS Port: /dev/ttyAMA0

Transmit Audio Source: ○ Rear/Data ● Front/Mic

Mode: ○ None ○ USB ● Data/Pkt

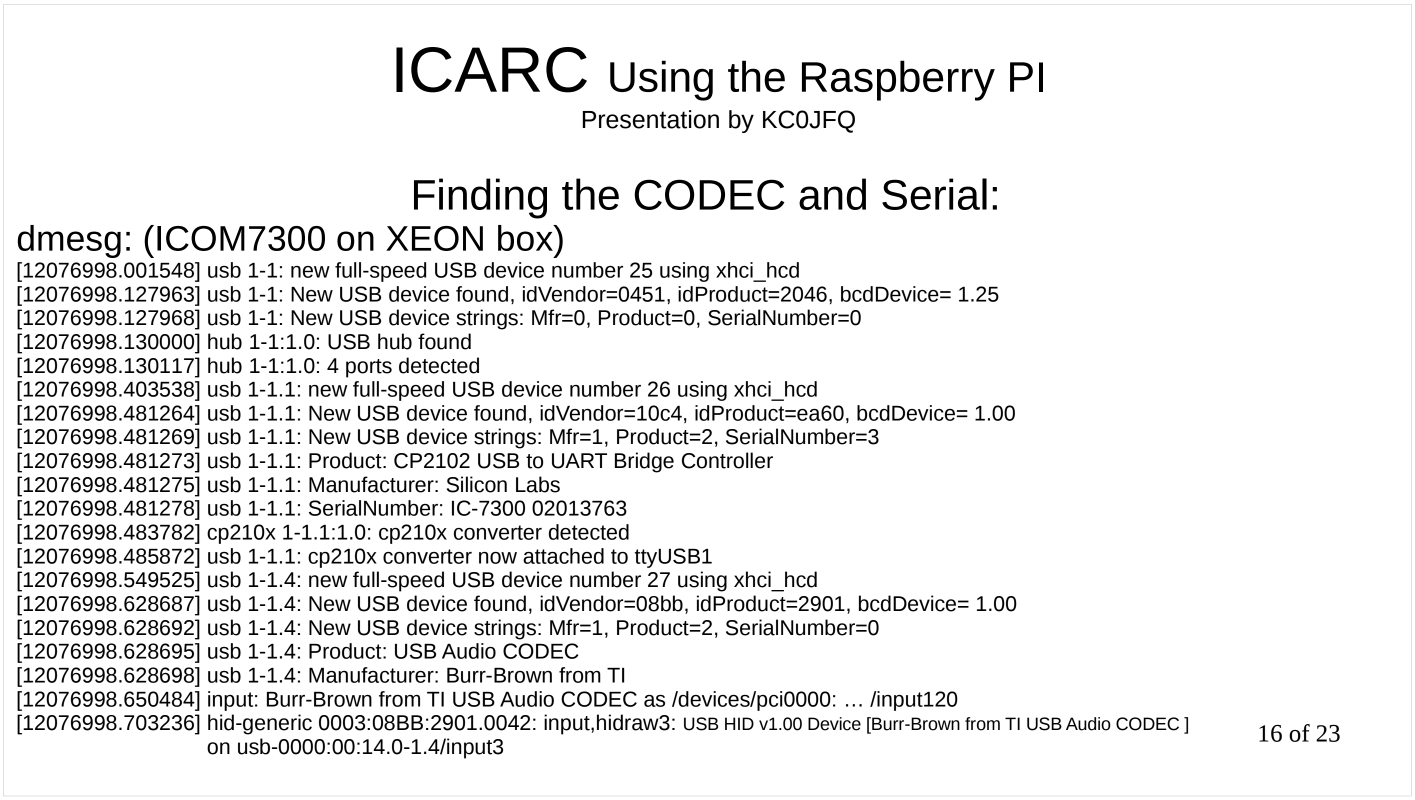
Split Operation: ● None ○ Rig ○ Fake It

Here is WSJTX from Monday morning.

Serial USB ports are quite mobile and WSJTX does a miserable job tracking them on both Linux and Windows.

Raspberry-PI has only the small set of devices needed to run the radio, so they aren't so *mobile*.

Audio tab: **CARD=CODEC,DEV=0** to select the PCM29xx device



ICARC Using the Raspberry PI

Presentation by KC0JFQ

Finding the CODEC and Serial:

dmesg: (ICOM7300 on XEON box)

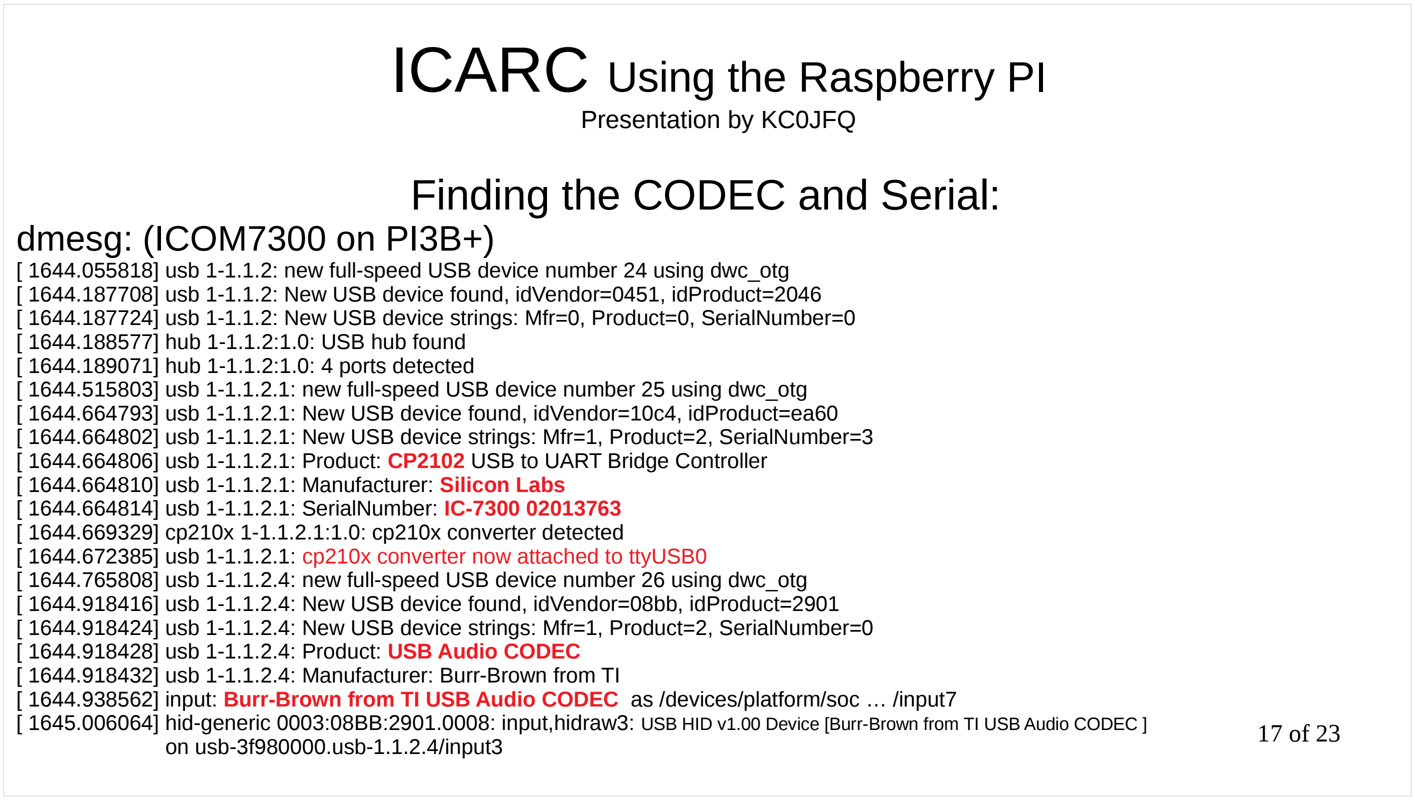
```
[12076998.001548] usb 1-1: new full-speed USB device number 25 using xhci_hcd
[12076998.127963] usb 1-1: New USB device found, idVendor=0451, idProduct=2046, bcdDevice= 1.25
[12076998.127968] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[12076998.130000] hub 1-1:1.0: USB hub found
[12076998.130117] hub 1-1:1.0: 4 ports detected
[12076998.403538] usb 1-1.1: new full-speed USB device number 26 using xhci_hcd
[12076998.481264] usb 1-1.1: New USB device found, idVendor=10c4, idProduct=ea60, bcdDevice= 1.00
[12076998.481269] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[12076998.481273] usb 1-1.1: Product: CP2102 USB to UART Bridge Controller
[12076998.481275] usb 1-1.1: Manufacturer: Silicon Labs
[12076998.481278] usb 1-1.1: SerialNumber: IC-7300 02013763
[12076998.483782] cp210x 1-1.1:1.0: cp210x converter detected
[12076998.485872] usb 1-1.1: cp210x converter now attached to ttyUSB1
[12076998.549525] usb 1-1.4: new full-speed USB device number 27 using xhci_hcd
[12076998.628687] usb 1-1.4: New USB device found, idVendor=08bb, idProduct=2901, bcdDevice= 1.00
[12076998.628692] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[12076998.628695] usb 1-1.4: Product: USB Audio CODEC
[12076998.628698] usb 1-1.4: Manufacturer: Burr-Brown from TI
[12076998.650484] input: Burr-Brown from TI USB Audio CODEC as /devices/pci0000: … /input120
[12076998.703236] hid-generic 0003:08BB:2901.0042: input,hidraw3: USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC ]
                  on usb-0000:00:14.0-1.4/input3
```

16 of 23

Here is what I look for to see what’s happening. These are low-level reports from the kernel. NO SUGARCOATING and no making up easy to understand terms that are all different (and impossible to remember).

These are on my big machine, an 8-core XEON. I plugged the 7300 in, captured the **dmesg** traffic and was done.

Lets move to the PI3B+ and do the saem...



ICARC Using the Raspberry PI

Presentation by KC0JFQ

Finding the CODEC and Serial:

dmesg: (ICOM7300 on PI3B+)

```
[ 1644.055818] usb 1-1.1.2: new full-speed USB device number 24 using dwc_otg
[ 1644.187708] usb 1-1.1.2: New USB device found, idVendor=0451, idProduct=2046
[ 1644.187724] usb 1-1.1.2: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[ 1644.188577] hub 1-1.1.2:1.0: USB hub found
[ 1644.189071] hub 1-1.1.2:1.0: 4 ports detected
[ 1644.515803] usb 1-1.1.2.1: new full-speed USB device number 25 using dwc_otg
[ 1644.664793] usb 1-1.1.2.1: New USB device found, idVendor=10c4, idProduct=ea60
[ 1644.664802] usb 1-1.1.2.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1644.664806] usb 1-1.1.2.1: Product: CP2102 USB to UART Bridge Controller
[ 1644.664810] usb 1-1.1.2.1: Manufacturer: Silicon Labs
[ 1644.664814] usb 1-1.1.2.1: SerialNumber: IC-7300 02013763
[ 1644.669329] cp210x 1-1.1.2.1:1.0: cp210x converter detected
[ 1644.672385] usb 1-1.1.2.1: cp210x converter now attached to ttyUSB0
[ 1644.765808] usb 1-1.1.2.4: new full-speed USB device number 26 using dwc_otg
[ 1644.918416] usb 1-1.1.2.4: New USB device found, idVendor=08bb, idProduct=2901
[ 1644.918424] usb 1-1.1.2.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1644.918428] usb 1-1.1.2.4: Product: USB Audio CODEC
[ 1644.918432] usb 1-1.1.2.4: Manufacturer: Burr-Brown from TI
[ 1644.938562] input: Burr-Brown from TI USB Audio CODEC  as /devices/platform/soc … /input7
[ 1645.006064] hid-generic 0003:08BB:2901.0008: input,hidraw3: USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC ]
               on usb-3f980000.usb-1.1.2.4/input3
```

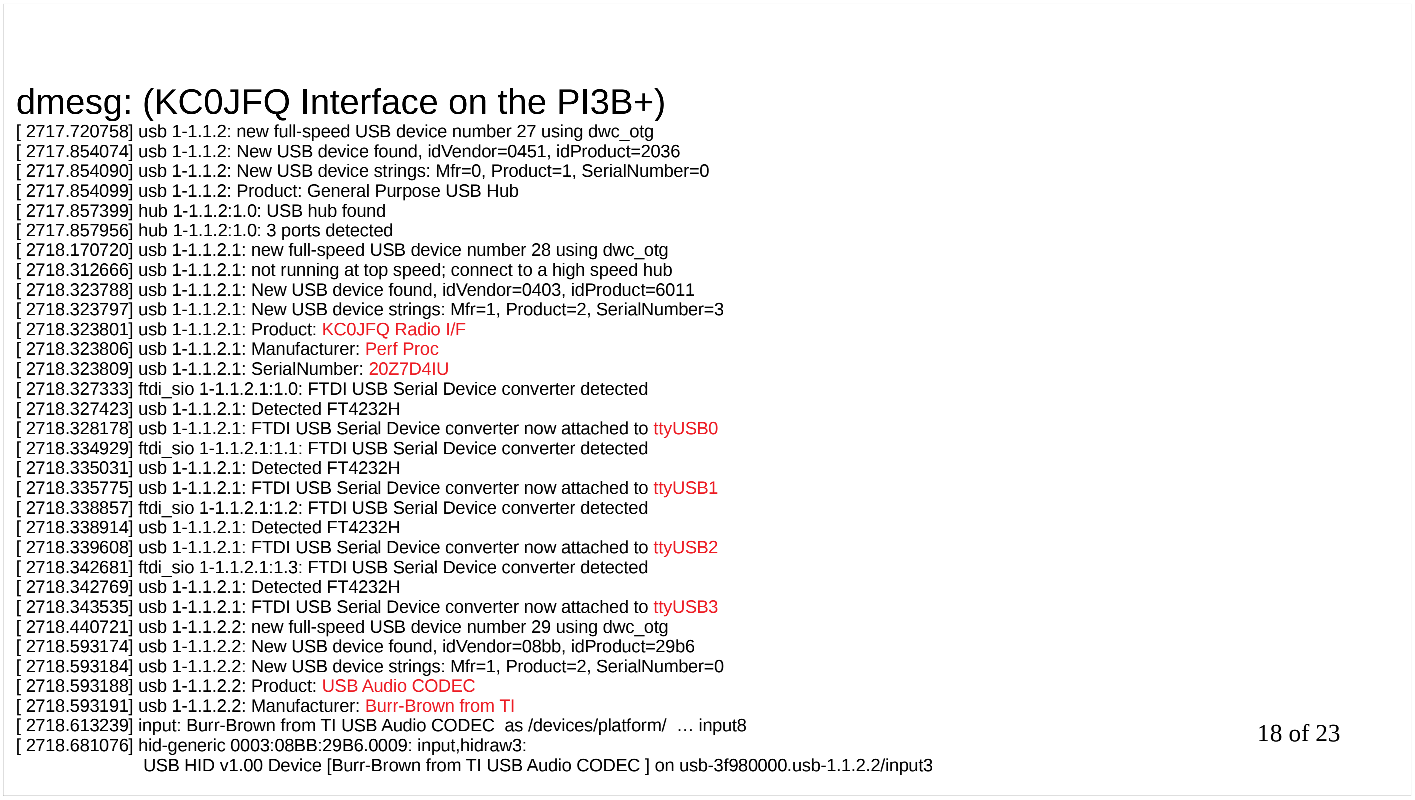
Here is **dmesg** from the PI3B+…

The IC7300 (and everyone else, for that matter) uses that stoopid CP2102. Its probably dirt cheap ‘cause so many are counterfeit. Follow the driver install procedure to the letter on your Windows Box. On Linux, just plug it in and it works.

The “**IC-7300 02013763**“ strings are stored in the CP2102.   **dmesg** tells you which port (/dev/TTYUSB0) the CP2102 is assigned to.

The PCM2906, which is a ancient BB design (Burr-Brown got absorbed by TI in 2000 for $7.6E9). Everybody uses it because it is the basis for most USB AUDIO drivers. It just works everywhere!

That “CODEC” mnemonic is what you’ll use in WSJTX and FLDIGI to connect audio.



dmesg: (KC0JFQ Interface on the PI3B+)

```
[ 2717.720758] usb 1-1.1.2: new full-speed USB device number 27 using dwc_otg
[ 2717.854074] usb 1-1.1.2: New USB device found, idVendor=0451, idProduct=2036
[ 2717.854090] usb 1-1.1.2: New USB device strings: Mfr=0, Product=1, SerialNumber=0
[ 2717.854099] usb 1-1.1.2: Product: General Purpose USB Hub
[ 2717.857399] hub 1-1.1.2:1.0: USB hub found
[ 2717.857956] hub 1-1.1.2:1.0: 3 ports detected
[ 2718.170720] usb 1-1.1.2.1: new full-speed USB device number 28 using dwc_otg
[ 2718.312666] usb 1-1.1.2.1: not running at top speed; connect to a high speed hub
[ 2718.323788] usb 1-1.1.2.1: New USB device found, idVendor=0403, idProduct=6011
[ 2718.323797] usb 1-1.1.2.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 2718.323801] usb 1-1.1.2.1: Product: KC0JFQ Radio I/F
[ 2718.323806] usb 1-1.1.2.1: Manufacturer: Perf Proc
[ 2718.323809] usb 1-1.1.2.1: SerialNumber: 20Z7D4IU
[ 2718.327333] ftdi_sio 1-1.1.2.1:1.0: FTDI USB Serial Device converter detected
[ 2718.327423] usb 1-1.1.2.1: Detected FT4232H
[ 2718.328178] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB0
[ 2718.334929] ftdi_sio 1-1.1.2.1:1.1: FTDI USB Serial Device converter detected
[ 2718.335031] usb 1-1.1.2.1: Detected FT4232H
[ 2718.335775] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB1
[ 2718.338857] ftdi_sio 1-1.1.2.1:1.2: FTDI USB Serial Device converter detected
[ 2718.338914] usb 1-1.1.2.1: Detected FT4232H
[ 2718.339608] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB2
[ 2718.342681] ftdi_sio 1-1.1.2.1:1.3: FTDI USB Serial Device converter detected
[ 2718.342769] usb 1-1.1.2.1: Detected FT4232H
[ 2718.343535] usb 1-1.1.2.1: FTDI USB Serial Device converter now attached to ttyUSB3
[ 2718.440721] usb 1-1.1.2.2: new full-speed USB device number 29 using dwc_otg
[ 2718.593174] usb 1-1.1.2.2: New USB device found, idVendor=08bb, idProduct=29b6
[ 2718.593184] usb 1-1.1.2.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 2718.593188] usb 1-1.1.2.2: Product: USB Audio CODEC
[ 2718.593191] usb 1-1.1.2.2: Manufacturer: Burr-Brown from TI
[ 2718.613239] input: Burr-Brown from TI USB Audio CODEC  as /devices/platform/  … input8
[ 2718.681076] hid-generic 0003:08BB:29B6.0009: input,hidraw3:
                 USB HID v1.00 Device [Burr-Brown from TI USB Audio CODEC ] on usb-3f980000.usb-1.1.2.2/input3
```

Here is **dmesg** from the PI3B+ when I plug my interface in…

This device uses the FTDIchip FT4232 4-channel USB-serial device. 4 channels for the price of 1! The radio is one channel and the Z8 processor that controls gain levels is on a second. A third channel goes to a DE9 for the Kenwood world.

You have to know which serial is which, but the 1.2.3.4 order is fixed, the CI-V channel is always the first one in the FT4232 group. The CI-V interface is electrically isolated from the radio as are the audio channels.

Finding which devices to use is pretty much the same as when connecting direct to the radio.

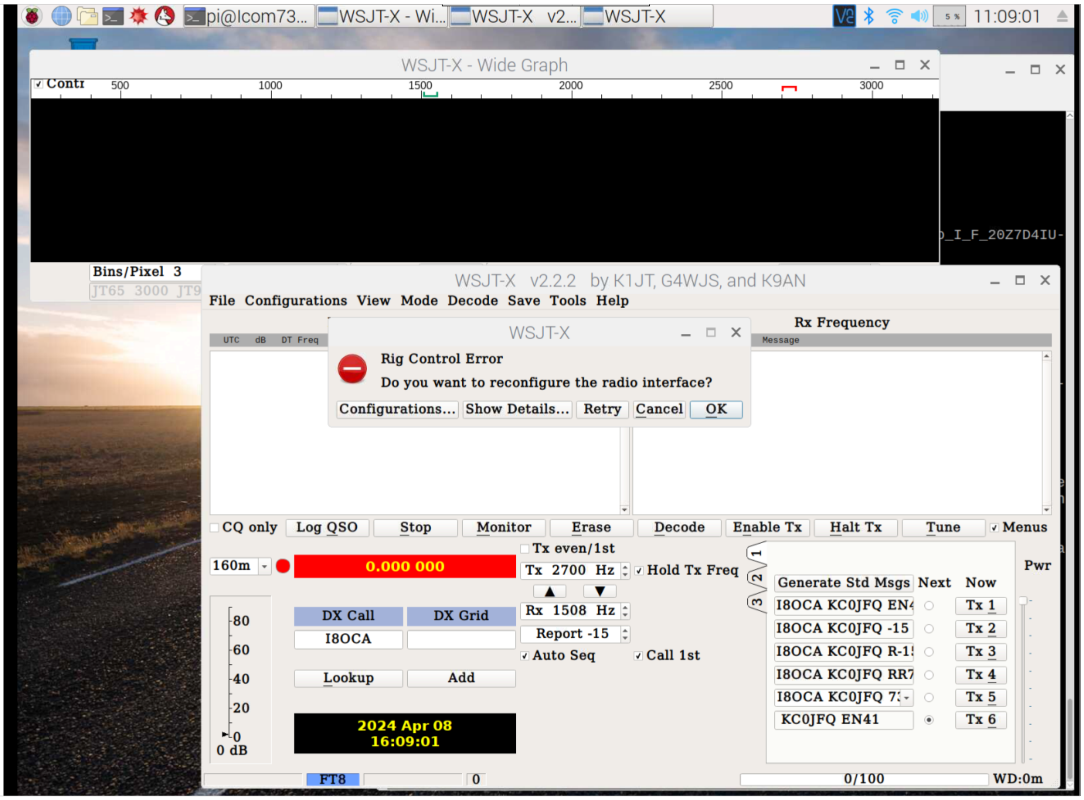
ICARC

Using the Raspberry PI

Presentation
by KC0JFQ
16

WSJTX

Operating without a radio connected



WSJT-X - Wide Graph

WSJT-X v2.2.2 by K1JT, G4WJS, and K9AN

WSJT-X

Rig Control Error

Do you want to reconfigure the radio interface?

Generate Std Msgs	Next	Now
I8OCA KC0JFQ EN4	○	Tx 1
I8OCA KC0JFQ -15	○	Tx 2
I8OCA KC0JFQ R-1	○	Tx 3
I8OCA KC0JFQ RR7	○	Tx 4
I8OCA KC0JFQ 73	○	Tx 5
KC0JFQ EN41	●	Tx 6

160m
0.000 000
DX Call: I8OCA
DX Grid:
Tx 2700 Hz
Rx 1508 Hz
Report -15
☑ Hold Tx Freq
☑ Auto Seq
☑ Call 1st
□ Tx even/1st
□ CQ only
☑ Menus
2024 Apr 08
16:09:01
FT8
0/100
WD:0m

Doesn’t like not being able to communicate with the radio :-(

We’ll try this on a real radio as you watch this presentation.

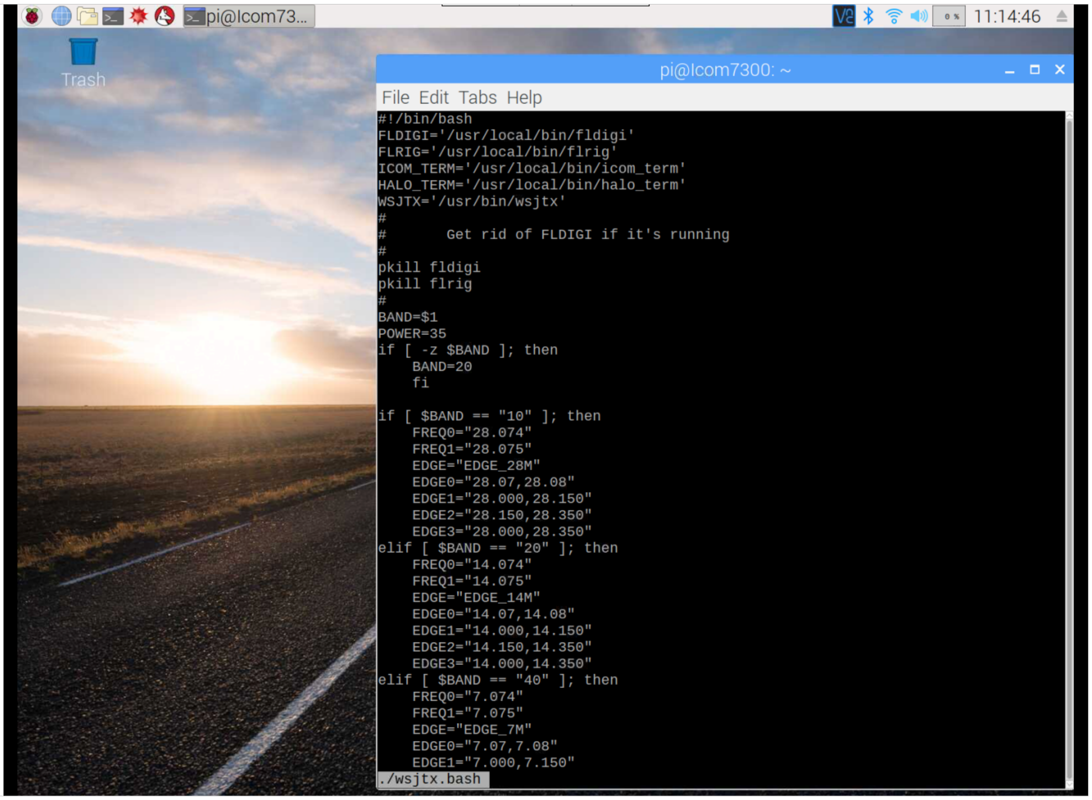
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 17

Starting WSJTX

Shell script



```
#!/bin/bash
FLDIGI='/usr/local/bin/fldigi'
FLRIG='/usr/local/bin/flrig'
ICOM_TERM='/usr/local/bin/icom_term'
HALO_TERM='/usr/local/bin/halo_term'
WSJTX='/usr/bin/wsjtx'
#
#       Get rid of FLDIGI if it's running
#
pkill fldigi
pkill flrig
#
BAND=$1
POWER=35
if [ -z $BAND ]; then
    BAND=20
    fi

if [ $BAND == "10" ]; then
    FREQ0="28.074"
    FREQ1="28.075"
    EDGE="EDGE_28M"
    EDGE0="28.07,28.08"
    EDGE1="28.000,28.150"
    EDGE2="28.150,28.350"
    EDGE3="28.000,28.350"
elif [ $BAND == "20" ]; then
    FREQ0="14.074"
    FREQ1="14.075"
    EDGE="EDGE_14M"
    EDGE0="14.07,14.08"
    EDGE1="14.000,14.150"
    EDGE2="14.150,14.350"
    EDGE3="14.000,14.350"
elif [ $BAND == "40" ]; then
    FREQ0="7.074"
    FREQ1="7.075"
    EDGE="EDGE_7M"
    EDGE0="7.07,7.08"
    EDGE1="7.000,7.150"
./wsjtx.bash
```

The shell script is 96 lines long, you give it the operating band and it sets the IC7300 spectrogram accordingly.

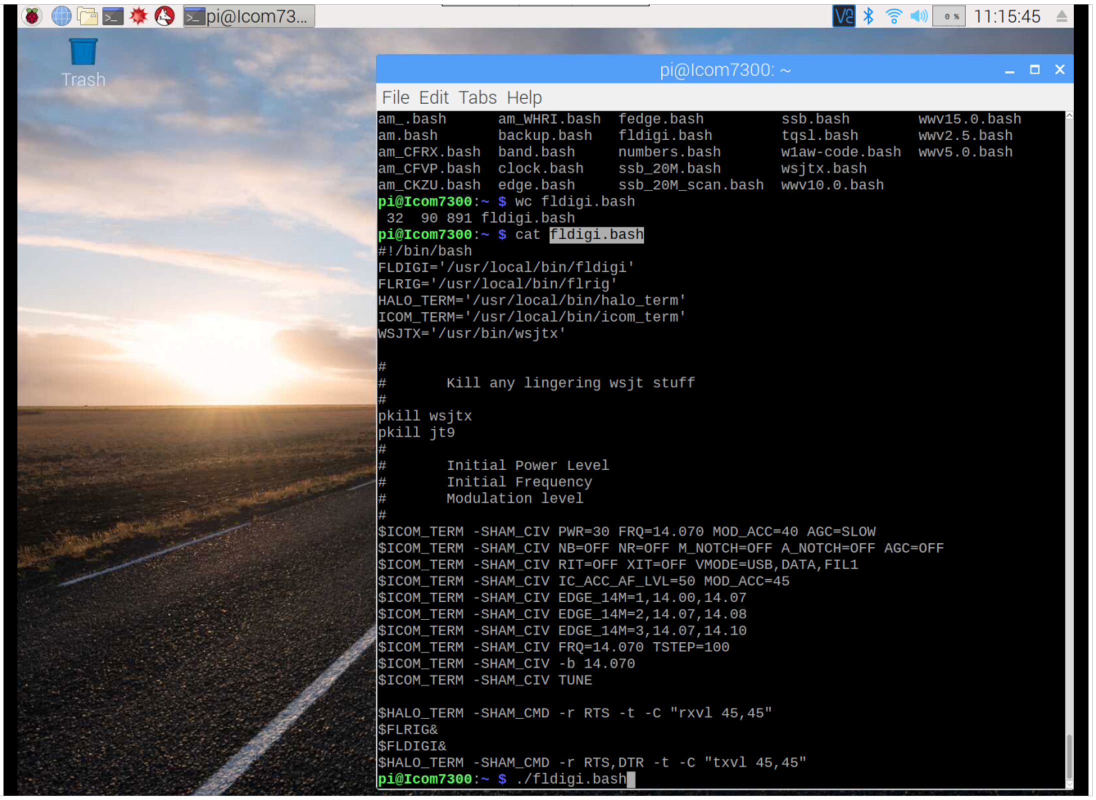
ICARC

Using the Raspberry PI

Presentation
by KC0JFQ
18

Starting FLDIGI (PSK31)

Shell script



```
am_.bash      am_WHRI.bash  fedge.bash         ssb.bash        wwv15.0.bash
am.bash       backup.bash   fldigi.bash        tqsl.bash       wwv2.5.bash
am_CFRX.bash  band.bash     numbers.bash       w1aw-code.bash  wwv5.0.bash
am_CFVP.bash  clock.bash    ssb_20M.bash       wsjtx.bash
am_CKZU.bash  edge.bash     ssb_20M_scan.bash  wwv10.0.bash
pi@Icom7300:~ $ wc fldigi.bash
 32  90 891 fldigi.bash
pi@Icom7300:~ $ cat fldigi.bash
#!/bin/bash
FLDIGI='/usr/local/bin/fldigi'
FLRIG='/usr/local/bin/flrig'
HALO_TERM='/usr/local/bin/halo_term'
ICOM_TERM='/usr/local/bin/icom_term'
WSJTX='/usr/bin/wsjtx'

#
#       Kill any lingering wsjt stuff
#
pkill wsjtx
pkill jt9
#
#       Initial Power Level
#       Initial Frequency
#       Modulation level
#
$ICOM_TERM -SHAM_CIV PWR=30 FRQ=14.070 MOD_ACC=40 AGC=SLOW
$ICOM_TERM -SHAM_CIV NB=OFF NR=OFF M_NOTCH=OFF A_NOTCH=OFF AGC=OFF
$ICOM_TERM -SHAM_CIV RIT=OFF XIT=OFF VMODE=USB,DATA,FIL1
$ICOM_TERM -SHAM_CIV IC_ACC_AF_LVL=50 MOD_ACC=45
$ICOM_TERM -SHAM_CIV EDGE_14M=1,14.00,14.07
$ICOM_TERM -SHAM_CIV EDGE_14M=2,14.07,14.08
$ICOM_TERM -SHAM_CIV EDGE_14M=3,14.07,14.10
$ICOM_TERM -SHAM_CIV FRQ=14.070 TSTEP=100
$ICOM_TERM -SHAM_CIV -b 14.070
$ICOM_TERM -SHAM_CIV TUNE

$HALO_TERM -SHAM_CMD -r RTS -t -C "rxvl 45,45"
$FLRIG&
$FLDIGI&
$HALO_TERM -SHAM_CMD -r RTS,DTR -t -C "txvl 45,45"
pi@Icom7300:~ $ ./fldigi.bash
```

Here is the FLDIGI script.

It doesn’t get the IC7300 neatly configured like the wsjtx does as I wasn’t running FLDIGI that much when I built this system.

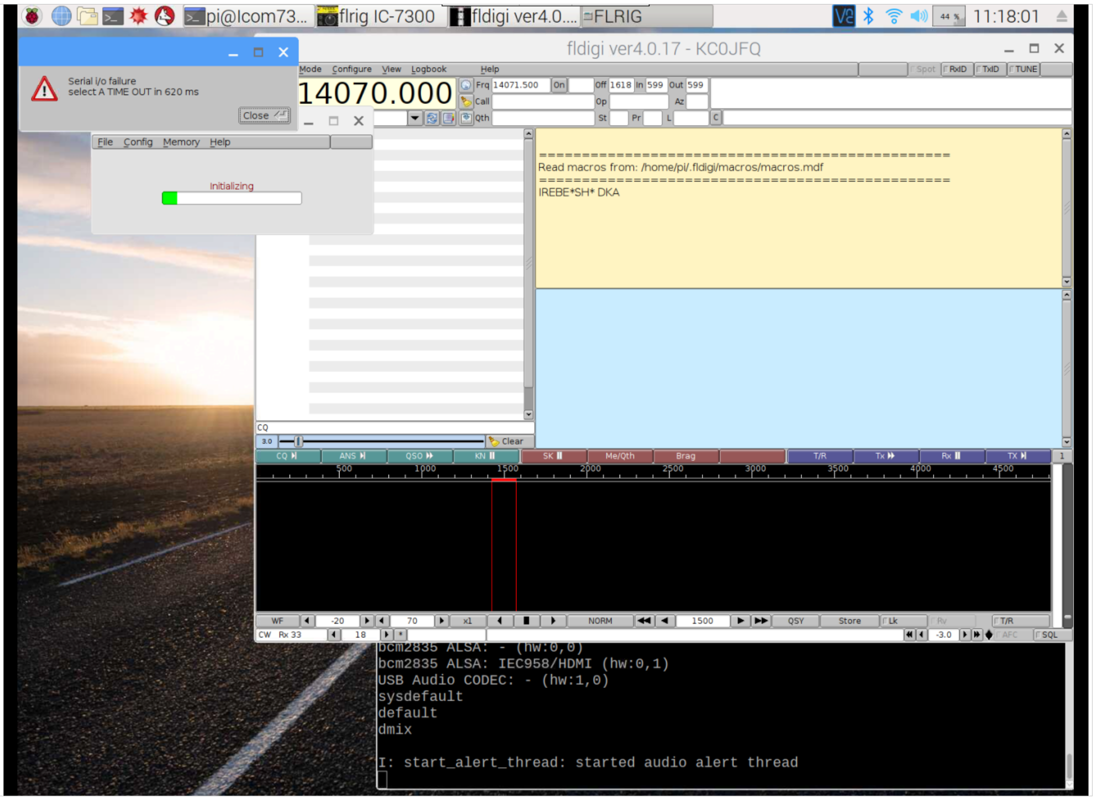
ICARC

Using the Raspberry PI

Presentation by KC0JFQ 19

Running FLDIGI

No radio error message



Serial i/o failure
select A TIME OUT in 620 ms

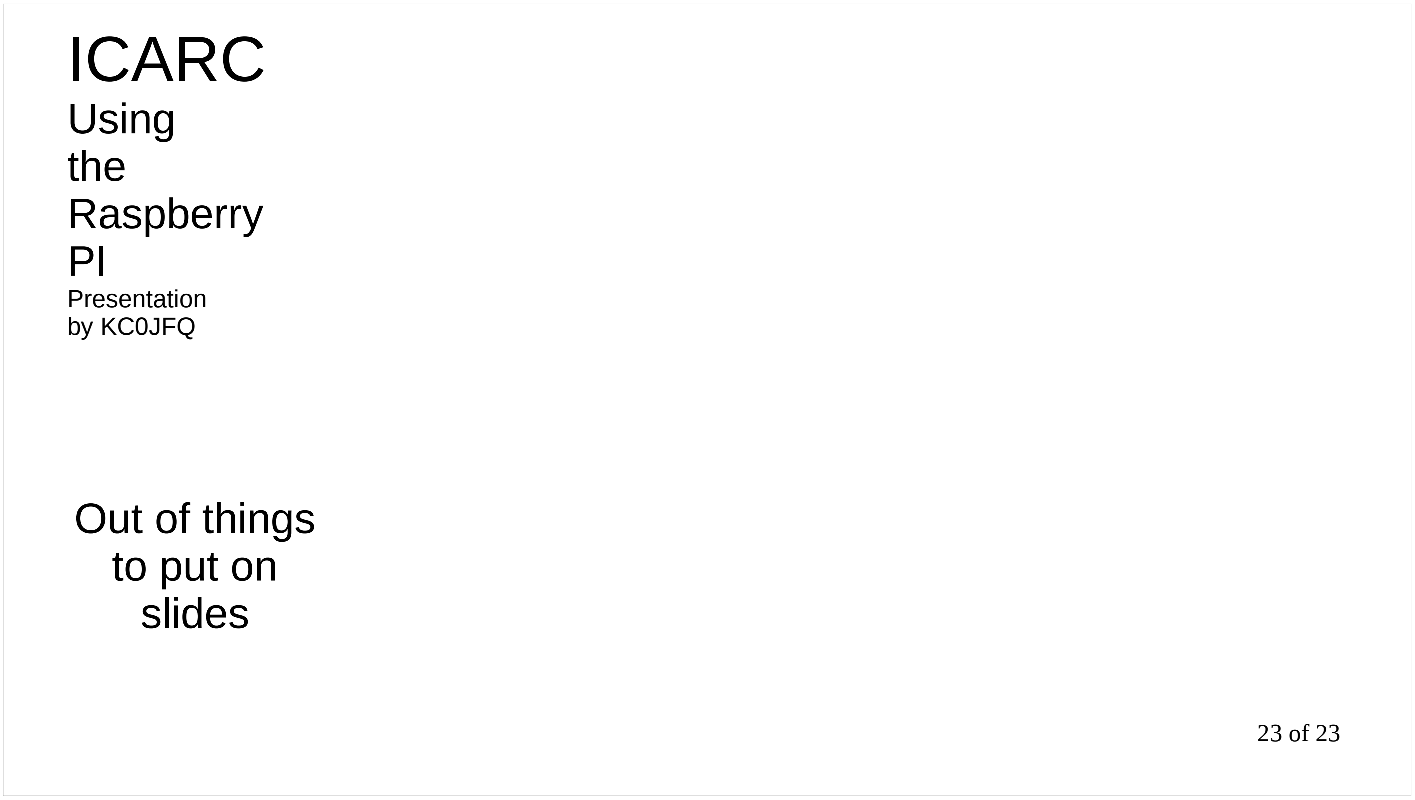
14070.000

Frq 14071.500 On Off 1618 In 599 Out 599
Call Op Az
Qth St Pr L C

```
==============================================
Read macros from: /home/pi/.fldigi/macros/macros.mdf
==============================================
IREBE*SH* DKA
```

```
bcm2835 ALSA: - (hw:0,0)
bcm2835 ALSA: IEC958/HDMI (hw:0,1)
USB Audio CODEC: - (hw:1,0)
sysdefault
default
dmix

I: start_alert_thread: started audio alert thread
```

Here is FLDIGI trying to operate without a radio.



ICARC

Using the Raspberry PI

Presentation by KC0JFQ

Out of things to put on slides

Here we be done!