Introduction – the ‘Hydra’

- How well does it work?
- Why I need many heads
- How it is configured
- RDP versus VNC

Introduction – Software using Draw module

- BASIC (sic) multi-tasking app !SignalBox
  - Plan in 1994 was to display a diagram of a signal box which changed as levers moved.
  - Diagram was held in memory as Draw data
  - Routines to zoom in and scroll were written
  - By 2007 an Acorn & Windows version done
  - Other software then used this approach:
  - !CountDn - !Cat - !FamTree - !DrawDis

Hydra

- Three computers in one
  - Based on Compute Module 4/5
  - Primary head has USB, HDMI sockets
  - All heads have wired Ethernet & NVMe drive
  - Primary head uses VNC to control other heads - their desktop displayed in Avalanche window
  - All three computers run independently
  - One (overall) reset switch

Hydra

- Why do I need three heads?
  - Virtual Risc PC still very useful and cheap
    - I need a Windows desktop anyway
    - One keyboard, one mouse (only one of me!)
    - Browsing (Windows) programming (RISC OS)
  - Windows plus Titanium
    - Two keyboards and mice, two monitors
    - Can leave one running a task but takes more room
  - Hydra – can do it all in one place

Hydra

- So far this presentation has been displayed using PDF reader on the laptop
  - Zoom is sharing the 1920x1200 screen of the laptop showing the PDF viewer
  - Now to display it using Iris on the Hydra
  - I will first start RealVNC Viewer on the laptop
  - Once it appears I will share the 1200 x 800 screen of the Hydra – then start Iris

Hydra

- This presentation now shows individual pages of a PDF under RISC OS on the Hydra
- The RISC OS desktop (1280 x 800) is being shared using RealVNC Viewer on my Windows laptop which is also running a Zoom session.
- The desktop on the Hydra is 3440 x 1440 by default but that is too large to use in Zoom
- Hydra desktop is scaled to fit the portable's screen resolution.

Hydra

- All three heads start up on power up
  - #1. shows RISC OS running the ROD network stack with the disc drives of #2 and #3 in ClusterFS and ShareFS respectively.
  - #2. running Linux on a CM5 booting from NVME
  - #3. running RISC OS on a CM4 booting from SD
  - Programmes on #2 and #3 can be launched from the Avalanche window. Data can be exchanged using the filer window on #1 and a Linux mount which can be seen in ClusterFS.

Hydra

- There is no ‘Audio’ socket so audio from #1 appears at the monitor over HDMI and I have a speaker plugged in to the monitor.
- Audio from #2 appears at a separate speaker directly.
- In both cases the audio just mixes with my voice and is picked up from there for Zoom.
- All of this is set up to work ‘out of the box’

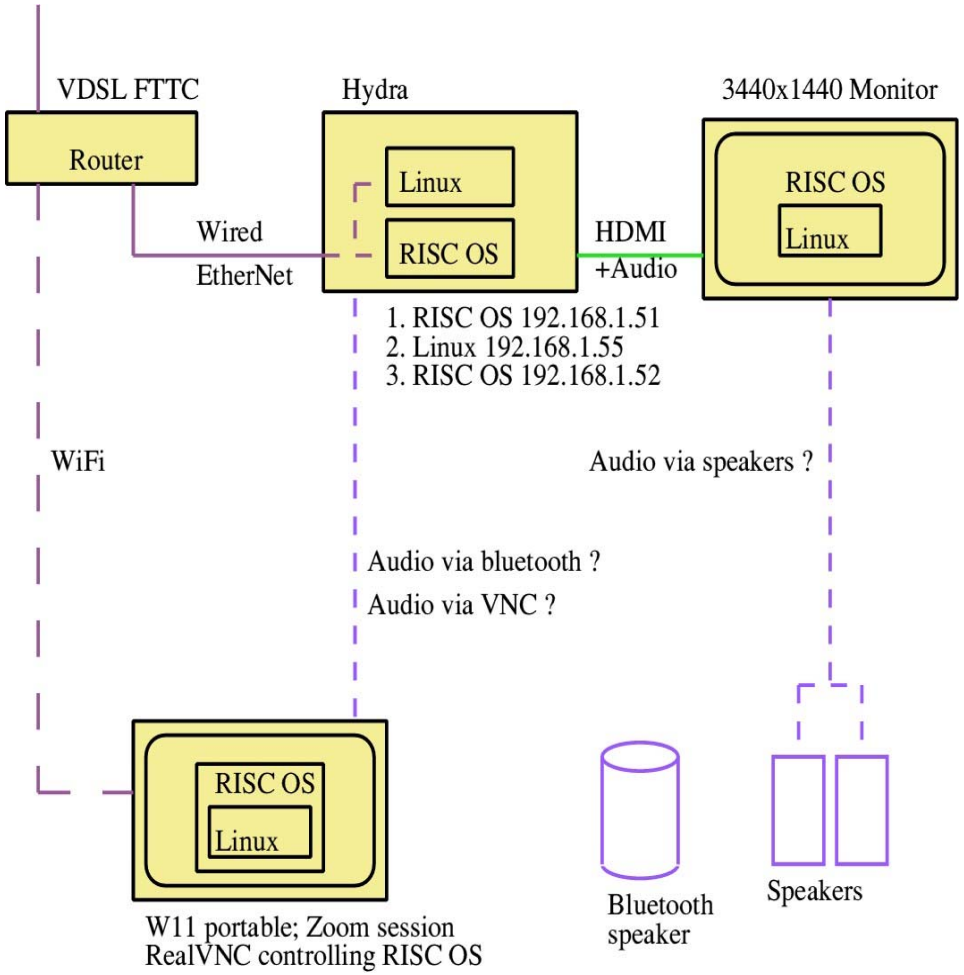
Hydra

- Example – I shall play a You-Tube video in a window which shows the Linux desktop
- Start Avalanche showing the Linux head
  - Start the browser and start a video
  - It will play in a small window updating at 16 fps
  - Audio will emerge from the speaker (camera)
  - If I play the video full screen, updates drop to about 1fps on the RISC OS desktop (RDP)
  - Via Zoom we have RDP over RDP over Zoom
  - Zoom audio is designed for speech not music

Hydra

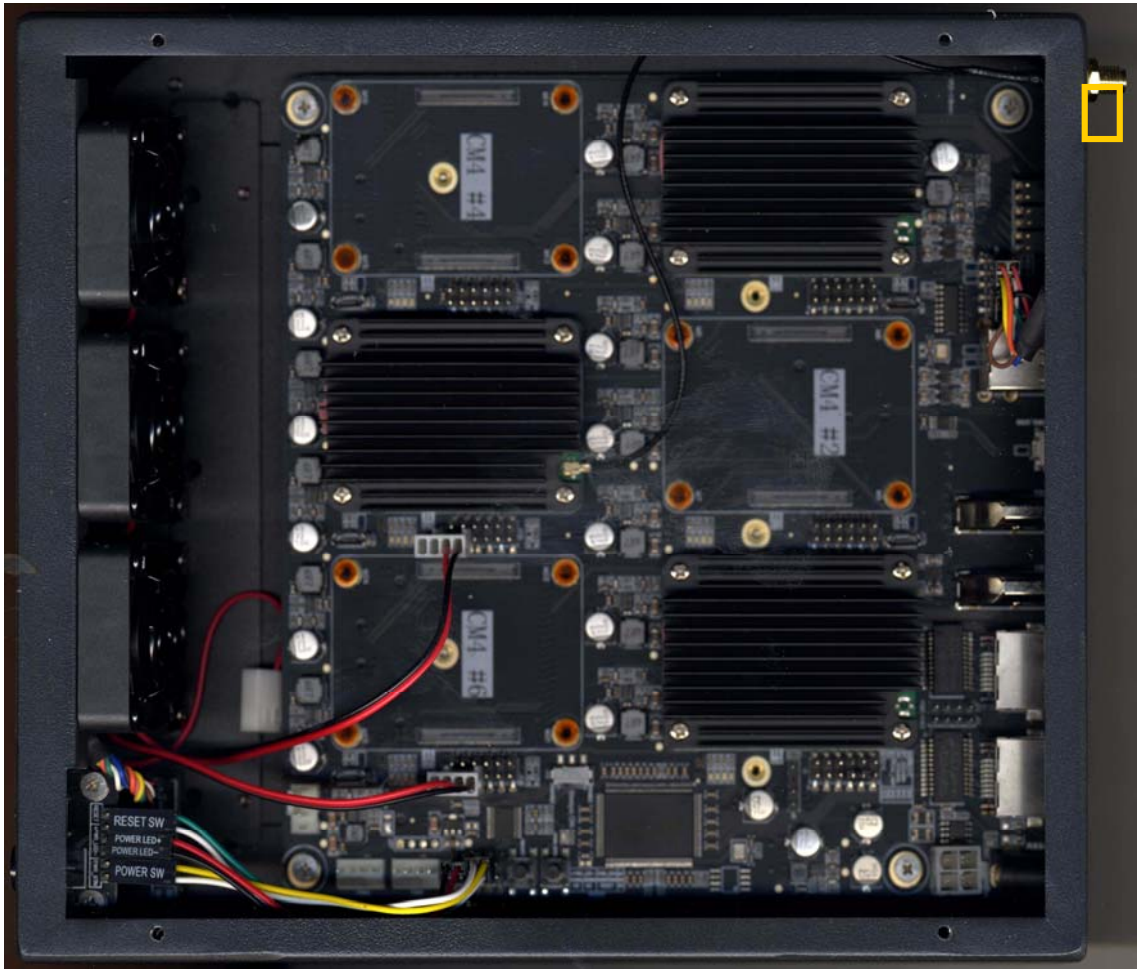
- The Linux head will be running whether or not an Avalanche window is open
- Start a Linux task & then close the Avalanche window, the task will continue to run.
- When the Linux task completes it can draw your attention by creating some sound output and placing the result in a shared folder
- You then look at the folder under RISC OS
- Tertiary RISC OS head similar (but no sound)

Hydra



- Hydra set-up
  - Shows Hydra & laptop displays

Hydra



- Looking inside:
  - Three heads
  - Three fans
  - No GPIO pins
  - (camera)

Hydra - summary

- Runs silently (fans are whisper quiet)
- Can take a few minutes to obtain net time
- Storage is fast (NVME drive uses 4k sectors)
- Well setup & configured to make it easy to use
- RISC OS under emulation in Linux is much less polished than Virtual Risc PC
- Support from R-Comp very helpful to iron out any gremlins

Hydra

- I have written software to control a real time clock, connected via USB
  - This ensures time is synchronised on booting.
- Digression on VNC versus RDP
  - RDP works best for multiple users with separate work spaces on a single resource
  - VNC transmits bitmaps from screen to screen and works best where there are different OS's – all users see the same screen
  - No RDP server ('remote' m/c) for RISC OS

Questions on Hydra

- I have a camera so I can show the Hydra close up
- Fire away!

Introduction – Software using Draw module

- BASIC (sic) multi-tasking app !SignalBox
  - Plan in 1994 was to display a diagram of a signal box which changed as levers moved.
  - Diagram was held in memory as Draw data
  - Routines to zoom in and scroll were written
  - By 2007 an Acorn & Windows version done
  - Other software then used this approach:
  - !CountDn - !Cat - !FamTree - !DrawDis

!SignalBox - basics

- Draw file data generated by MakeDraw and held in memory within application
  - Unlike a Draw file, you have to specify the end of data (as there is no EOF marker object)
  - Rendering can be step by step (using Draw_Fill and Draw_Stroke etc.) or use DrawFile_Render
  - Path objects with both outline and fill colours set to -1 are invisible
  - With objects at known locations, alternative shapes can be displayed by poking colours.

!SignalBox - basics

- Draw file data held in memory
  - Use Wimp_UpdateWindow if only a few, known objects have changed and render just those
  - Else use Wimp_RedrawWindow and re-render
  - General routines will now scale and translate the Draw data to fill the window on screen so that it acts as a ‘window’ on whole or part of the data
  - Specific routines to process mouse clicks and menu selections for the particular application
  - Conversion to run as a multi-tasking application under Windows (www.svrsig.org/software/SoftCon.htm).

!SignalBox - basics

- Example running !SignalBox under Windows
  - Runs on Windows95SE, 98, NT, Millenium, 2000, XP, 7, 10 and 11 (here Windows 11)
  - Start Signalbox.exe and show it running
  - As levers are pulled, the screen changes to show points and signals responding – this is done by poking the Draw data and redrawing the items
  - Can be downloaded from www.svrsig.org/software/SignalBox.zip

!SignalBox - basics

- How is signal box diagram created?
  - Use !Trace to generate vector graphic data showing the running lines, each section is linked to the next, just like plugging together two pieces of track. Links are conditional at a point or signal
  - Each section, point and signal is at a known memory location in the Draw data

Other applications

- Basic principle of rendering Draw file data
  - First extension was to display mechanical locking, building up the vector graphic by combining standard shapes but linking them so that movement of one item will cause corresponding movement in those connected to it
  - As items are moved, the screen changes to show other items responding – this is done by poking the Draw data and redrawing the items
  - Can be downloaded from www.svrsig.org/Lock.zip

!Locking

- First extension circa 2007
  - Uses generic routines to render Draw file data
  - Uses specific routines to move items by poking their bounding box and transformation matrix
- Shown running on RISC OS
- Data can be regenerated as all source data are provided
-

!CountDn

- Next extension circa 2012
  - Uses generic routines to render Draw file data
  - Uses specific routines to change text and paths being displayed by poking their contents
- Shown running on RISC OS
- Calculates a difficulty matrix for a tile selection in about 8cs using ARM code

!Cat

- Next extension circa 2014
  - Uses generic routines to render Draw file data
  - Uses specific routines to generate ‘tree’ diagram in memory by reading contents of selected disc drive
  - Allows data to be saved as a text file listing every file and path and then for offsets to be applied
  - Importing edited file automatically updates ‘tree’ display so that items are moved clear of each other

!FamTree

- Next extension circa 2018
  - Uses generic routines to render Draw file data
  - Uses specific routines to generate ‘tree’ diagram in memory by reading contents of selected disc drive
  - Allows data to be saved as a text file listing every file and path and then for offsets to be applied
  - Importing edited file automatically updates ‘tree’ display so that items are moved clear of each other
  - Family Tree is top down, not left to right

!DrawDis

- Next extension circa 2020
  - Uses generic routines to render Draw file data
  - Uses specific routines to identify draw object under mouse and show names of groups
  - Mouse click can turn a group invisible using Draw object &4C0 (invisible group)
  - Helps when editing a Draw file created elsewhere by identifying position in memory of an object
  - Disassembly of Draw file data (using !MultiTask) identifies memory location of each object

!RingBind

- Next extension 2022
  - Uses generic routines to render Draw file data
  - Uses specific routines to animate turned pages
  - Displays Draw files extracted from a PDF in an ‘open book’ format
  - Images held internally as a set of Draw file data containing each page as Draw file data with a tag object to hold values of internal pointers
  - Great fun to write but a bit of a gimmick

Editing a PDF

- Editing vector graphics in a PDF
  - Often the content of a PDF is vector graphic
  - !ArtWorks allows export of a Draw file
  - !MultiTask allows disassembly of the Draw file
  - !DrawDis identifies memory location of an object
  - !MakeDraw allows precise recreation of Draw file with e.g. changed colours or thickness of lines
  - PDF can be regenerated using level 3 Postscript printer drivers and ps2pdf13

Questions on my software

- Fire away!