Purpose

This document covers the design overview and build steps for the Hornet.

Back to 16/4/21 in validating discord
Complete GPS documentation – including Simconnect and how DCS doesn't get it yet – pyHWLink_SimConnect_Transmitter.py

Acknowledgements

The Hornet was built following early Beta Open Hornet designs. The Open Hornet project is nothing short of amazing, with full 3D cad models and electronic designs. Full credit to John (Noctum) and the team for delivering a truly open set of plans for the Hornet.

The bulk of communications occur on Discord.
https://discord.com/invite/G5PA5ju
https://openhornet.com/
https://github.com/jrsteensen/OpenHornet

The electronics used in this pit aren't based on the Open Hornet stack as they Open Hornet stack wasn't available at the beginning of the build.

Ben and Luke have been a big part of the team in building the Hornet in Oz. Thanks gents. Ben's repository is full of useful bits and pieces (https://github.com/Ben-F111/Ben-F18-Sim-Pit), additional info is found on his Facebook page - https://www.facebook.com/BenF18Sim/

Table of Contents

Systems Overview

The Hornet currently supports DCS and MSFS 2020. DCS provides full system capabilities, whilst MSFS 2020 supports a basic aircraft tooling. The bulk of the interfacing with DCS uses the Flight Panels fork of DCS-BIOS. Further details on API usage and development can be found in 'Python HW Link – Design' document.

The throttle and stick assembly are from WinWing (https://www.winwing.cn/en/JOYSTICK) and the rudder pedals are from MFG (https://mfg-sim.com/en/).

As the ADI is not natively rendered in DCS, Helios is used to generate the ADI. The Left and right DDIs along with the AMPCD can be easily exported, but views such the RWR/TEWS, do require a little more work to export.

Systems Design Approach

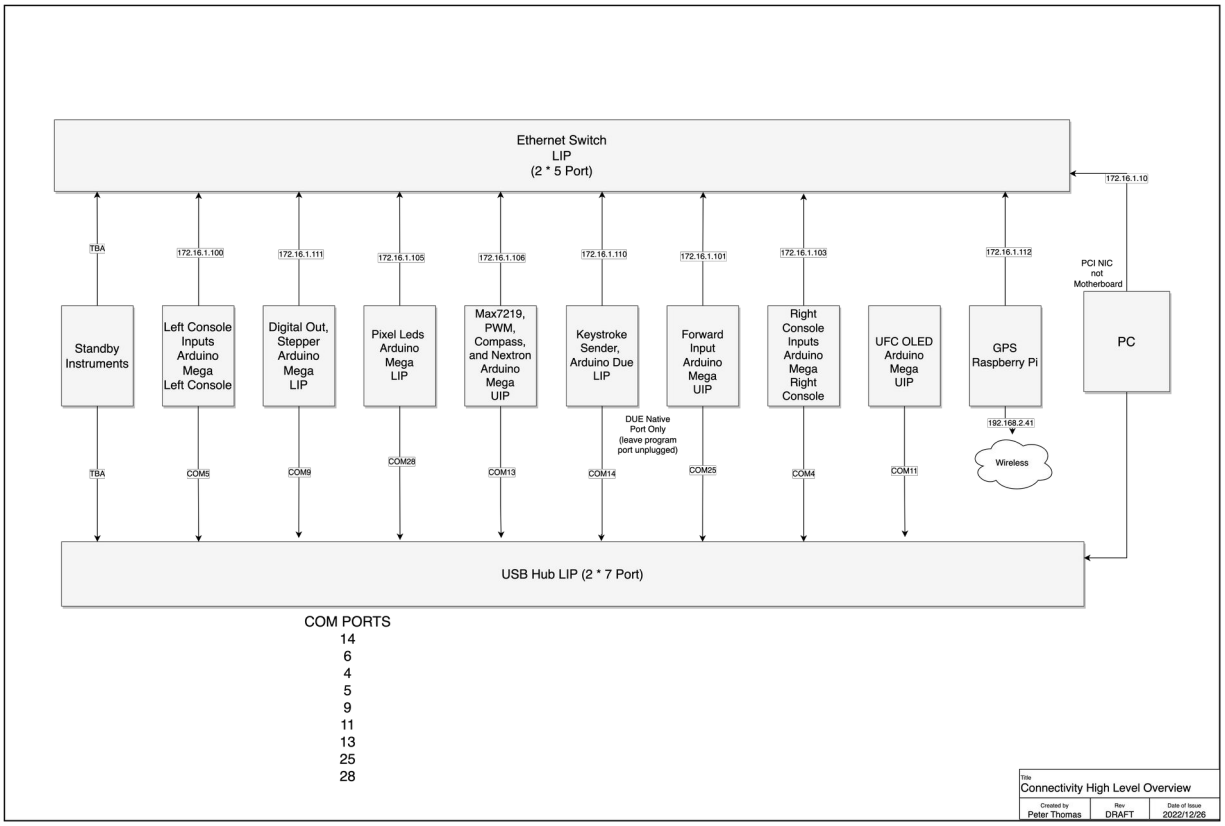
Where possible Off the Shelf building blocks and Open Source have been used to ensure longevity and adaptability of the pit. Code and Designs unique to the build (ie not Open Hornet components) are published in the Git Hub Repo (https://github.com/bnepethomas/bne-arduino-flight-simulator-interfaces). I'll acknowledge now the filing system in the repo needs tidying up.

The repo holds a number of current and historical projects, which makes finding things 'fun' at times. The bulk of the software is divided into two regions, DCS-BIOS and Python HW Link. References to locations for the project will be relative to the root directory of the repo.

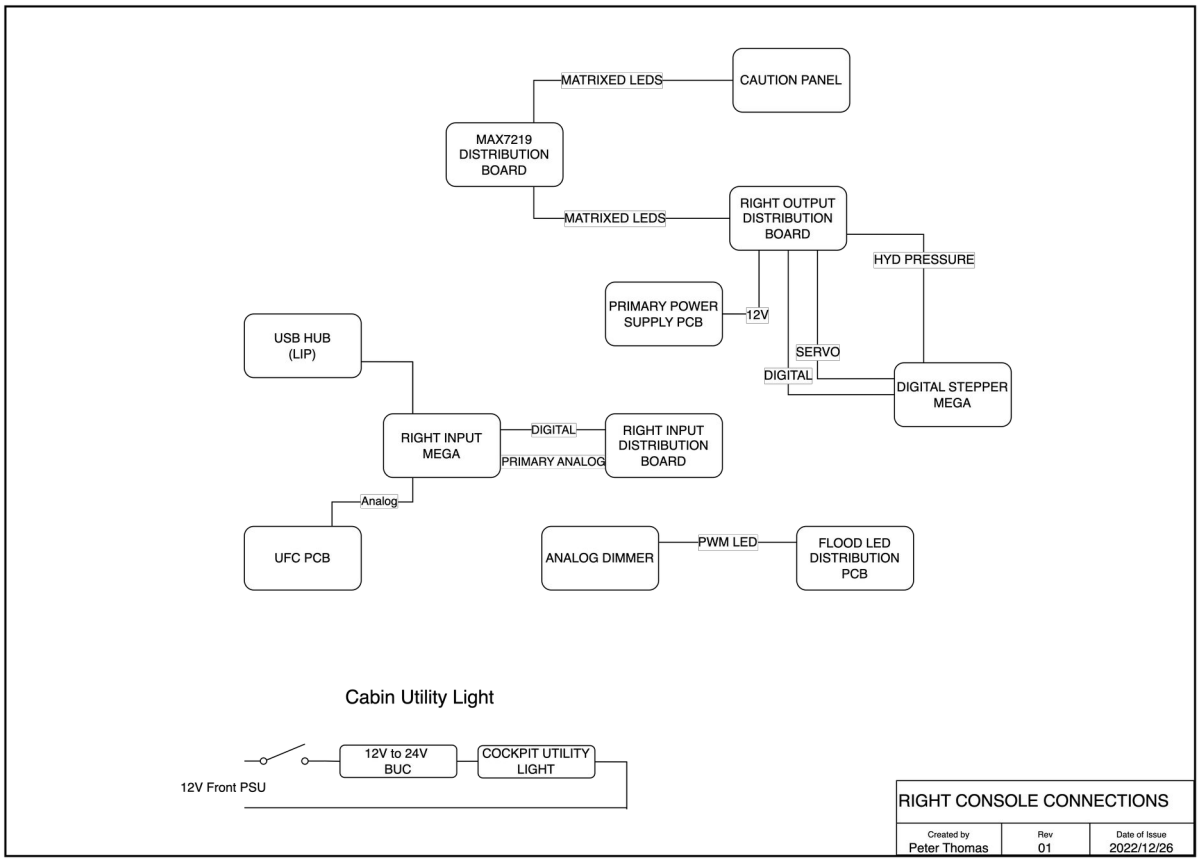
KICAD was used for PCB design. It is a fully integrated package which ensures a consistency between schematics and the PCB.

LibreOffice and Draw.io are used for documentation.

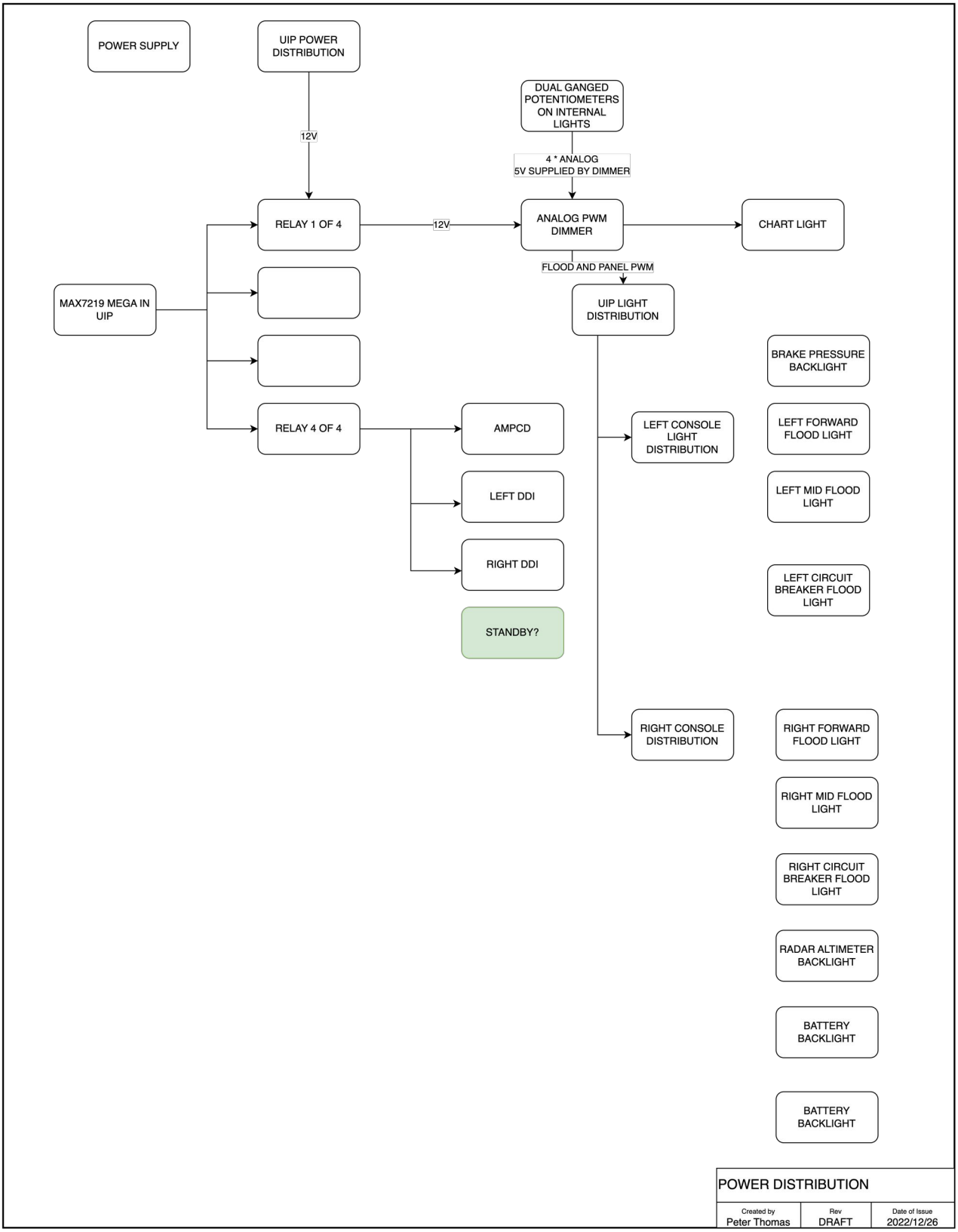
Arduino Mapping



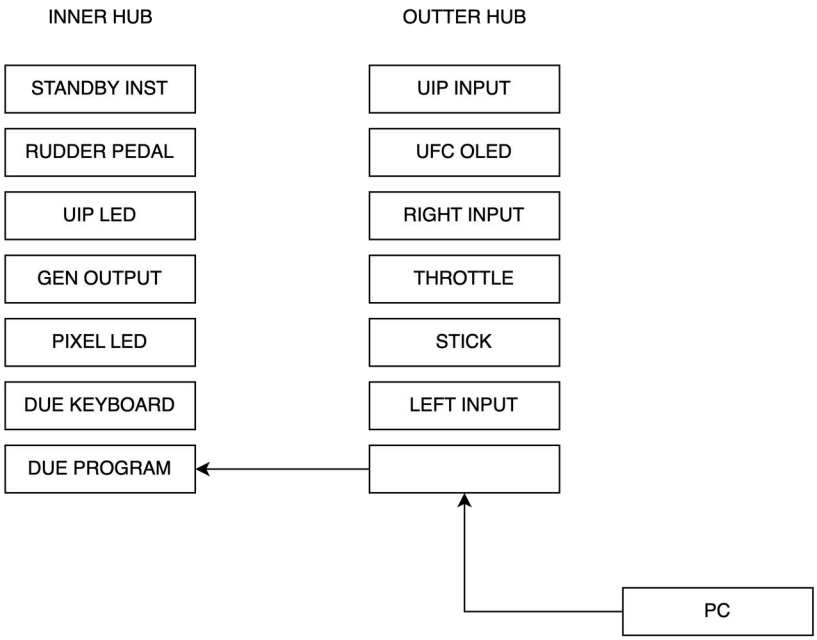
Right Console

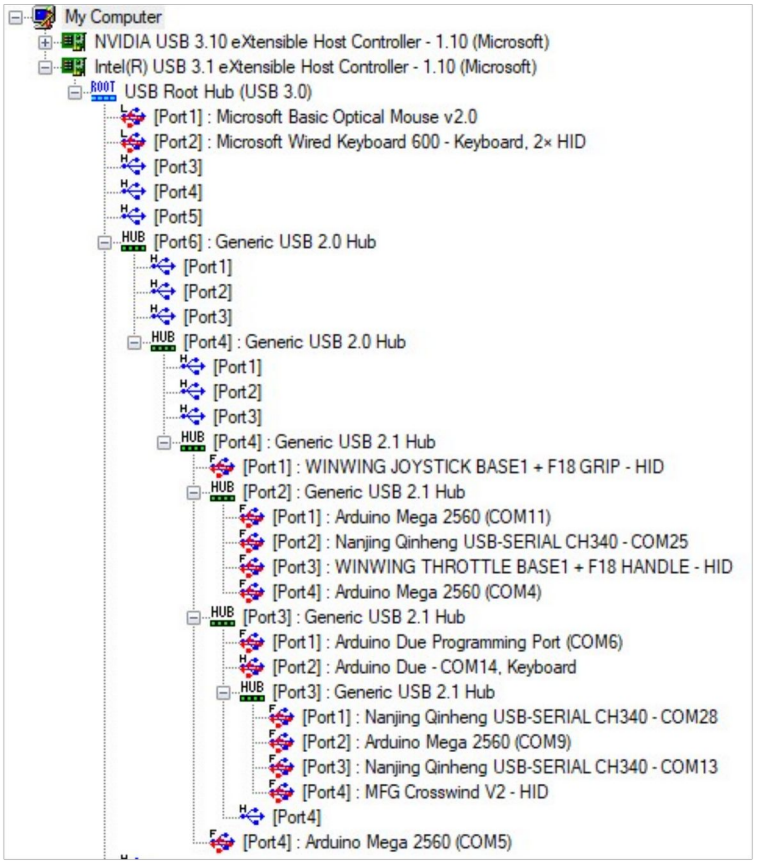


Power Distribution



USB DETAILS





Title		
USB CABLING		
Created by Peter Thomas	Rev 01	Date of Issue 2022/12/26

Hardware Design Approach

Whilst the ultimate goal is use as little USB connectivity as possible for simulator to pit communications, the bulk of the Arduinos are currently connected with USB and Ethernet.

Switch and Analog Inputs

The bulk of the Switch inputs are multiplexed to an Arduino Mega. The input Megas are located in the Left and Right Consoles as well as the UIP. A small PCB takes the ports 22 to 53 to an Expansion PCB that is specific to the area it is serving. This approach enables code to be easily shared and reduces the amount of bespoke troubleshooting. The current exception to this is the Standby Instrument inputs, which now have a dedicated Mega which deals with both inputs and outputs on the Standby Instrument Panel.

The of the panels has a panel specific PCB that connects all services required for that panel aside from backlighting, as the backlighting PCB has its own connectors. This panel holds the diodes used to prevent ghosting of the switch inputs.

The bulk of the analog inputs are connected to the input Megas using the same connectivity approach as switch inputs. A small capacitor is used to reduce the amount of electrical noise received on the Mega. One exception to this is the approach used to dim the AOA indicators. The AOA dimmer sends updates over Ethernet to the Mega responsible for driving the LEDs. The original PCB design used a small connector to bring the Mega Analog inputs up to the primary analog input PCB. Later releases extended the lower PCB to include the analog inputs with a header bringing the analog inputs to the riser.

OLED, LED, Coil Outputs, Addressable RGB LEDs

The UFC displays are OLEDs driven using an I2C Multiplexor. Originally the Adafruit libraries where used, but ultimately found the U2G library offered the greatest performance and flexibility.

Almost all of the LEDs in the pit are driven using Max7219s. As the Max7219 does not support individual LED dimming, LED outputs are grouped into regions, enabling individual dimming of the region. This means that some of the possible outputs are not used. All of the Max7219s are located on a single PCB and are daisy chained. LED outputs (which are matrixed), are then either connected to the assembly (such as the AoA) or connected to the output distribution PCBs in the left and right console.

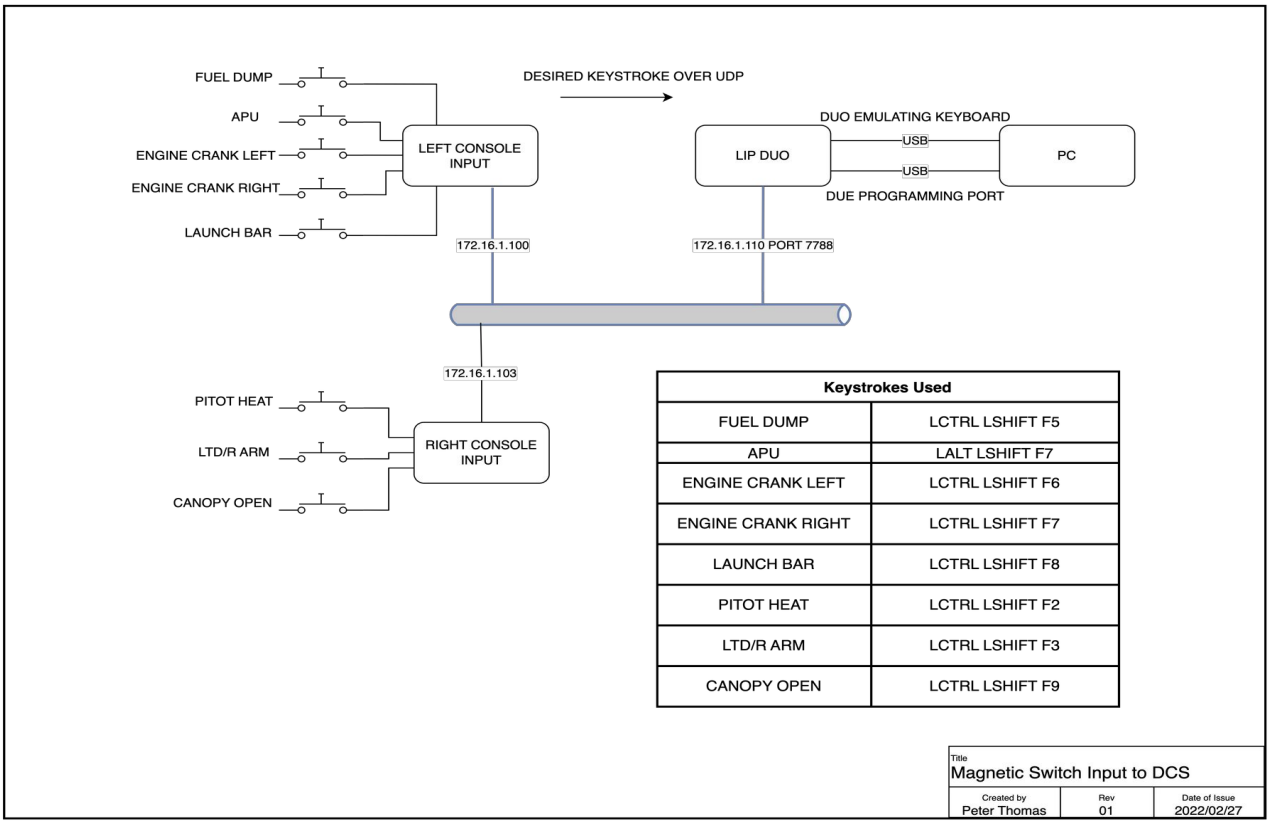
Magnetic Coil outputs use UN2003 Darlington Drivers arrays. This are located on output distribution PCBs in the Left and Right consoles.

Addressable RGB LEDs (WS2812) are used for backlighting. There are design options for AoA, but the final build used LEDs driven by Max7219.

DCS Systems Design Approach

DCS and Magnetic Switches

Currently the DCS F18 module doesn't provide the same level of support as the A10. If the DCS-BIOS interface is used to set a switch position, the simulator thinks the switch is being physically held in position. This prevents it being automatically released. Because of this, a different approach is needed to support the magnetic switches. Instead of using DCS-BIOS a message is sent from the input Mega over Ethernet to an Arduino Due. The Due in turn sends keystroke to the PC.



Other Software Modules

Flight Panels DCS-BIOS Fork

As the original author [FSF]Ian of DCS-BIOS is no longer maintaining DCS-BIOS, another team is maintaining a fork of it. Two of the DCS Flightpanels repositories are used in this project. One is the Arduino library, and the other is the PC side which sends the data streams to the Arduinos over USB. Whilst there is mention of Ethernet, specially multicast, it does not appear be heavily used.

https://github.com/DCSFlightpanels/dcs-bios

https://github.com/DCSFlightpanels/dcs-bios-arduino-library

Unlike the original DCS-BIOS, Flightpanel's DCS-BIOS does support resuming a flight after edits, which seriously speeds up coding and troubleshooting as flights no longer need to be exited and restarted.

The PC monitoring GUI is currently a plug-in for the Chrome browser, once you've worked out how to install and activate it, the plug-in is a very useful monitoring tool.

SOCAT is used to send data to Arduinos, a 32 and 64 bit is included, requiring you to copy the desired version to the folder above.

It is possible to create a shortcut of the CMD to the task bar, but it does require a bit of goofing around, you can't simply drag a shortcut of the CMD file to the taskbar. The steps needed to achieve this ar

Helios

Helios, among other things, renders instrument and updates values using a LUA feed. Helios is now only needed for the ADI as the other steam instruments are no longer using a screen.

Helios itself

https://github.com/HeliosVirtualCockpit/Helios/releases

Helios F18 profiles

http://www.captzeen.com/helios/f18.asp

Fly Elise

Fly Elise provides both design and operate capabilities for curved screens. The curved screen was designed using Immersive Designer PRO. Design Pro is able to generate monitor configuration files used by DCS, and hopefully MSFS 2020 in the future.

I'm still yet to get to the bottom of why I see image drift when using the projectors. It could be due to the following:

- Slight movement in the ceiling
- Temperature changes in the projector itself
- The projector reverting to automatic keystone correction. Generally for the simulator it is recommended that automatic keystone correction is disabled.

Explicitly setting hot-keys should be done for Fly Elise, as the default mappings may conflict with the default settings in the setting in DCS or MSFS 2020.

Currently the multi-view capabilities in MSFS aren't geometrically accurate, so support is currently not provided in Fly Elise (as of October 2020).

When running DCS you will get a very unusual experience when you enter the flight. Its important not to enable simulator specific capabilities until the flight has started. Once the flight is loaded the screen will look very distorted, but once the simulator capabilities are enabled the view will come good.

Blog Updates regarding MSFS 2020

https://fly.elise-ng.net/blog/microsoft-fligh-simulator/ - August 22, 2020

Our latest Immersive Display PRO v4.3.0 supports warping and blending of Microsoft Flight Simulator 2020, both for Standalone Edition and for Steam Edition.

However, there is a catch. Although this initial version of MSFS supports multiple outputs and multiple monitors, it does NOT support multiple windows (at first). So you will be able to use several monitors for a single view but you won't be able to open multiple views or use a 2nd window for instruments.

This is a huge limitation for being able to use MSFS for true geometrical outside views in a multi-projection or multi-LCD systems. Only a single view (spanned displays) can be used with the limitation of the maximal FOV that can be achieved with a single view, which is about 100 degrees. Larger FOVs will cause the so-called pinch distortion.

In order to get 100% accurate and geometrically correct images, our Immersive Calibration PRO calculates the needed view parameters (FOV and orientation) to produce perfect and undistorted views with no FOV limitation. But , in order to be able to use those views, MSFS must support the so-called multi-view or multi-window functionality.

Unfortunately, at this moment, MSFS does NOT support this, and multi-view setups are not possible.

https://fly.elise-ng.net/blog/multi-view-perspective-correction/ - July 16, 2021

Designing the projection setup for 100% accurate and true geometric correction for 3D game engines and simulation setups requires even more attention and design considerations. Computer generated graphics and perspective views have a limitation of the total field of view (FOV) that can be rendered to a single view. In practice, rendering FOVs of 100 or more degrees will result in a "pinch" distortion and unusable view. As a consequence, creating a visual projection setups with FOVs of 180 or larger horizontal FOV, requires a 3D game engine to generate multiple views. Each view will be generated with a FOV of less than 100 degrees and combined and edge by Immersive Display PRO.

Almost all flight simulators P3D, DCS, X-Plane as well as all graphics engines like Unity, Unreal, etc.. already support configurable multi-view rendering.

However, combining multiple perspective views as one continuous undistorted image on a curved projection screen requires additional image geometric correction (stretching and squeezing) which is dependent on the physical projection setup, projection screen, projectors light path and the designed eye-point.

Designing a projection setup for multi-view projection involves additional analysis for projector pixel utilization and image stretching from a design eyepoint.

The latest version of Immersive Designer PRO introduces a new analysis module for analysis and evaluation of the perspective image pixel utilization and stretching.

Component Details

Arduinos and Ethernet Shields

In earlier projects I have used clone Arduinos and Ethernet shields, had some real issues with reliability of the Ethernet shields. For the Hornet ran with genuine Arduino Megas and Ethernet Shields (v2). The Ethernet shields have been solid.

A benefit of using genuine Arduinos is they report a serial number, which becomes very useful when you have eight possible devices to program.

External Lights

The external lights (Position, Formation and Navigation) are driven from the MOSFET PWM PCB.

PWM dimming using Arduino driven MOSFETs are used with the desired output level being acquired from Flightpanel's DCS-BIOS. Of note is that not all Arduino Mega Digitals pins support PWM.

Flood Lighting

Flood lights are distributed throughout the pit. They are close as possible to the original components. All have been printed using PLA+ on the Prusa. A Cold White filament was used as it offers great reflectivity. Led Bulbs and bulb holders where used to reduce the amount of heat.

PWM dimming using Arduino driven MOSFETs are used with the desired output level being acquired from Flightpanel's DCS-BIOS. Of note is that not all Arduino Mega Digitals pins support PWM

GPS

The GPS is not native to the Hornet, but is helpful for navigation and approaches. It is driven by both DCS and MSFS 2020. The Lowrance Air Map 2000C is capable of receiving NEMA (National Marine Electronics Association) Sentences over an RS232 interface, which means it can be driven by an external source such as a high resolution GPS or a simulator.

In other Sims I've used FSUIPC and C# applications to acquire Lat Long information. As the Hornet project is a test bed for several new techniques and tools, a new approach was used for driven the GPS.

The GPS itself is connected to a Raspberry Pi with a simple PCB (Pi GPS Serial), (hosting a Max232 to convert 3.3V Logic levels to RS232.

The MSFS 2020 interface uses a Python module to interface to Sim Connect (https://github.com/bnepethomas/Python-SimConnect). This was forked off https://github.com/odwdinc/Python-SimConnect which at the time had an error which prevented it from working with MSFS2020. That error has since been resolved.

One learning is when updating Python Modules – The mod that has been install (for Windows) lives in c:\users\admin\Local\Programs\Python\Python311\Lib\site-packages\module-name. So if you are updating the module to add capabilities, don't forget to copy the modified module back to the git repo.

Left Console – FCS

The Rudder trim assembly has a press to Trim button which rotates the assembly to a take-off trim position. This is driven by a servo, which would normally prevent the pilot from manually turning the trim knob. To address this restriction, the servo is 'disconnected' in software when not being driven by the take-off trim action.

On startup the Trim knob is shifted just off centre, and then returned to centre after a short delay. The take-off trim action is sensed by the press of the Trim Button (onToTrimBtnChange)

```
Servo TRIM_servo;
#define TrimServoPin 12
#define TRIM_servo_Off_Center_Pos 800
#define TRIM_servo_Center_Pos     1450


void CenterTrimServo() {
  TRIM_servo.attach(TrimServoPin);
  TRIM_servo.writeMicroseconds(1100);  // set servo to "Mid Point"
  delay(10);
  TRIM_servo.writeMicroseconds(800);  // set servo to "Mid Point"
  delay(300);
  TRIM_servo.detach();
}
void onToTrimBtnChange(unsigned int newValue) {
  if (newValue == 1) {
```

```
    CenterTrimServo();
  }
}
DcsBios::IntegerBuffer toTrimBtnBuffer(0x74b4, 0x2000, 13, onToTrimBtnChange);
void setup() {
  TRIM_servo.attach(TrimServoPin);
  // set servo to just off Centre
  TRIM_servo.writeMicroseconds(TRIM_servo_Off_Center_Pos);

  delay(1000);
  // set servo to centre
  TRIM_servo.writeMicroseconds(TRIM_servo_Center_Pos);
  delay(500);
  TRIM_servo.detach();
}
```

To enable the pilot to turn the Rudder trim the servo The Rudder trim servo and potentiometer u

Panel Backlighting

The Panels are driven by a hardware solution developed by Luke. It uses PCBs with SMD (Surface Mounted) LEDs, which enable knobs to be effectively backlit, something that is difficult when using through hole LEDs.

The LEDs used are from the WS2812B family, who do not have a separate clocking or data load pin. This has the major benefit of enabling 100's of LEDs to be driven with a four wire interface (Power, Ground, Data-In, Data-Out). It does however introduce a timing consideration, the LEDs expect a constant data feed when updating, i.e. the Arduino library (FastLED) should not be interrupted when doing an update. If it is interrupted all sorts of strange colours and state is encountered, often toward the end of the led chain. As the Arduino DCS-BIOS library uses interrupts to receive serial data over the USB interface, and most Arduino CPUs are only single threaded the DCS-BIOS and FastLED are not good neighbours with long LED chains.

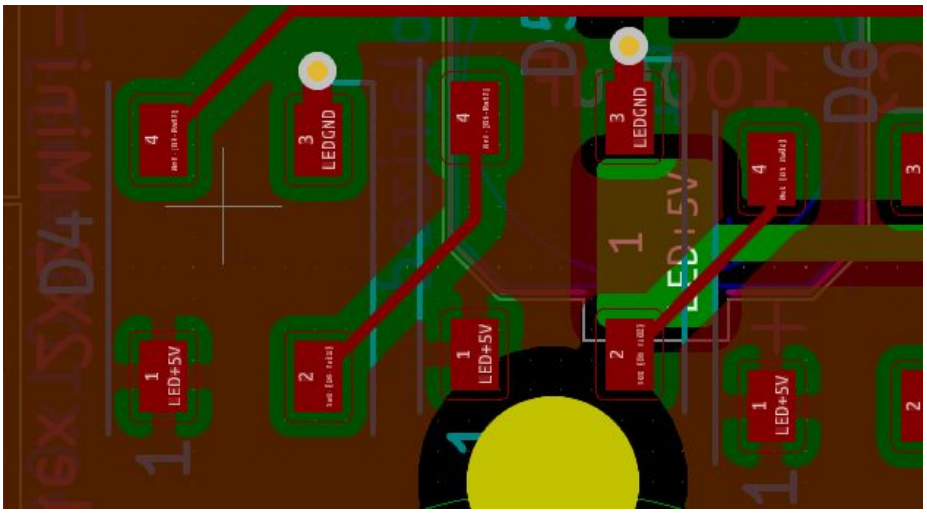
With a pretty full fleet of Pixel Leds (1500 in my test setup) - it takes 50mS to update them - so during that time an interrupt from serial comms through DCS BIOS is likely to cause a hiccup. Reduce the number of pixels down to 30 and the update time reduces to 3mS

This challenge was solved by splitting DCS-BIOS and the FastLED library onto two Arduino's with the DCS-BIOS Arduino sending lighting updates over Ethernet to the FastLED library.

Some devices are backlight using PWM, this includes some of the first generation of gauges. Later gauges (eg Standby Instruments) use the 5mm Through hole WS2812 LEDs.

The backlighting levels are derived from DCS-BIOS, the Flight Panels fork presents these values, whereas the original DCS-BIOS does not.

A design note when using ground and power planes with PCBs, Thermal Reliefs should be included as this reduces the amount of heat needed to solder or remove a device. For reference - https://resources.pcb.cadence.com/blog/2021-pcb-thermal-relief-guidelines-for-effective-layouts.



The 5mm NeoPixel LEDs used in the Standby instruments use a different colour encoding, From (https://www.adafruit.com/product/1938), ‘These are "RGB" instead of "GRB" format used in the 5050-sized LEDs you are so used to’.

Power Supplies and Power Management

A single PC power supply powers all devices in the pit. This includes the USB Hubs, Backlighting, and Monitors.

Some of the power rails are controlled by relays to give a full dark pit experience. Whilst there was concern about the monitors loosing their position on the PC – it appears to hold

The current orientation of the power supply is not ideal, with the fans bearings making a noise on initial power up. Ultimately it should be relocated or reorientated.

The original plan was to use different sized connectors for the 5V and 12V services. Despite the best efforts, still managed to mix them up, so in addition to attempting to use different size connectors, the following colour coding is used

5V	Grey
12V	Red

The Left Console, Right Console, UIP, and LIP have power distribution PCBs.

Rudder Pedals

The pedals are MFG Crosswinds, and are beautifully engineered. The optional Damper is well worth the investment. The footrests aren’t from MFG, at the time of purchase, a team on the DCS

forums was manufacturing F-16 footrests (https://forum.dcs.world/topic/256148-mfg-crosswind-f-16-combat-pedals/)

The pedals are mounted on a plate driven by a motor, enabling the pedals to be adjusted using a switch in the forward lower panel assembly. Limit switches are used. The plate is mounted on a pair of rails.

Throttle Assembly

As the WinWing throttle was not part of the original Open Hornet design the panels to the left and forward of the throttle required modification.

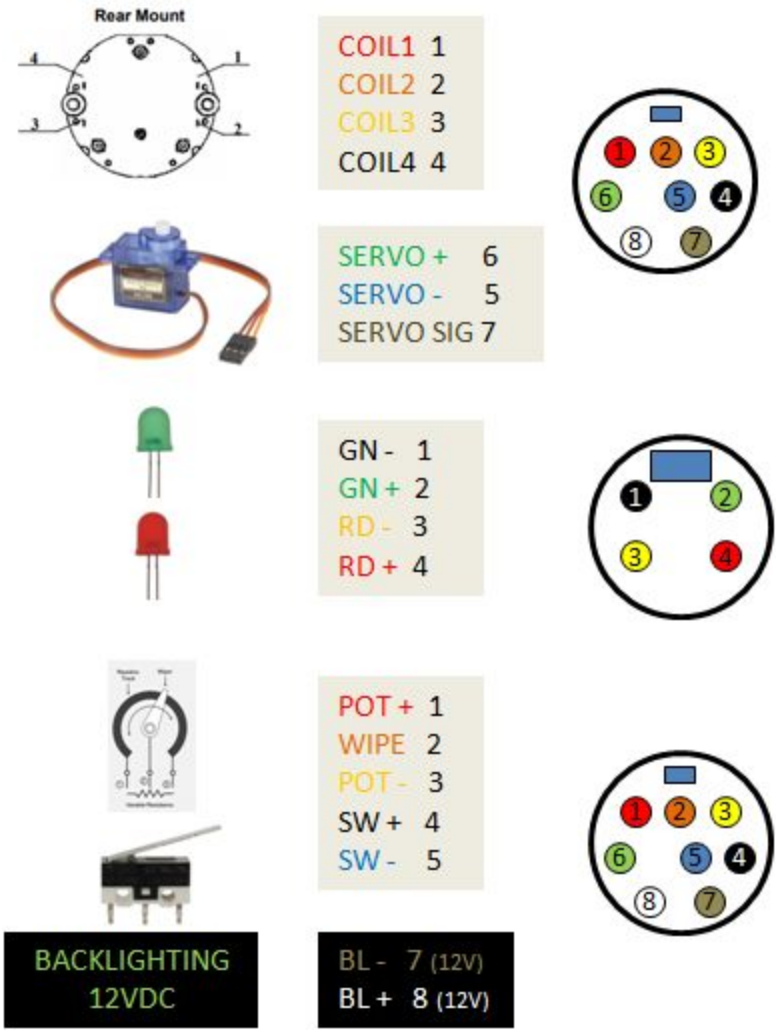
The throttle was removed from its outer enclosure, and the backlighting was modified, in that is not longer driven by the WinWing electronics, but instead is driven by the custom PWM (Pulse Width Modulation) dimmer. Additional resistors were added to allow the LEDs to be driven by 12V.

Right Console Utility Light

The light is a genuine Utility Light with a 24V bulb. It is driven by an inverter which has its power supply controlled by a near-by switch.

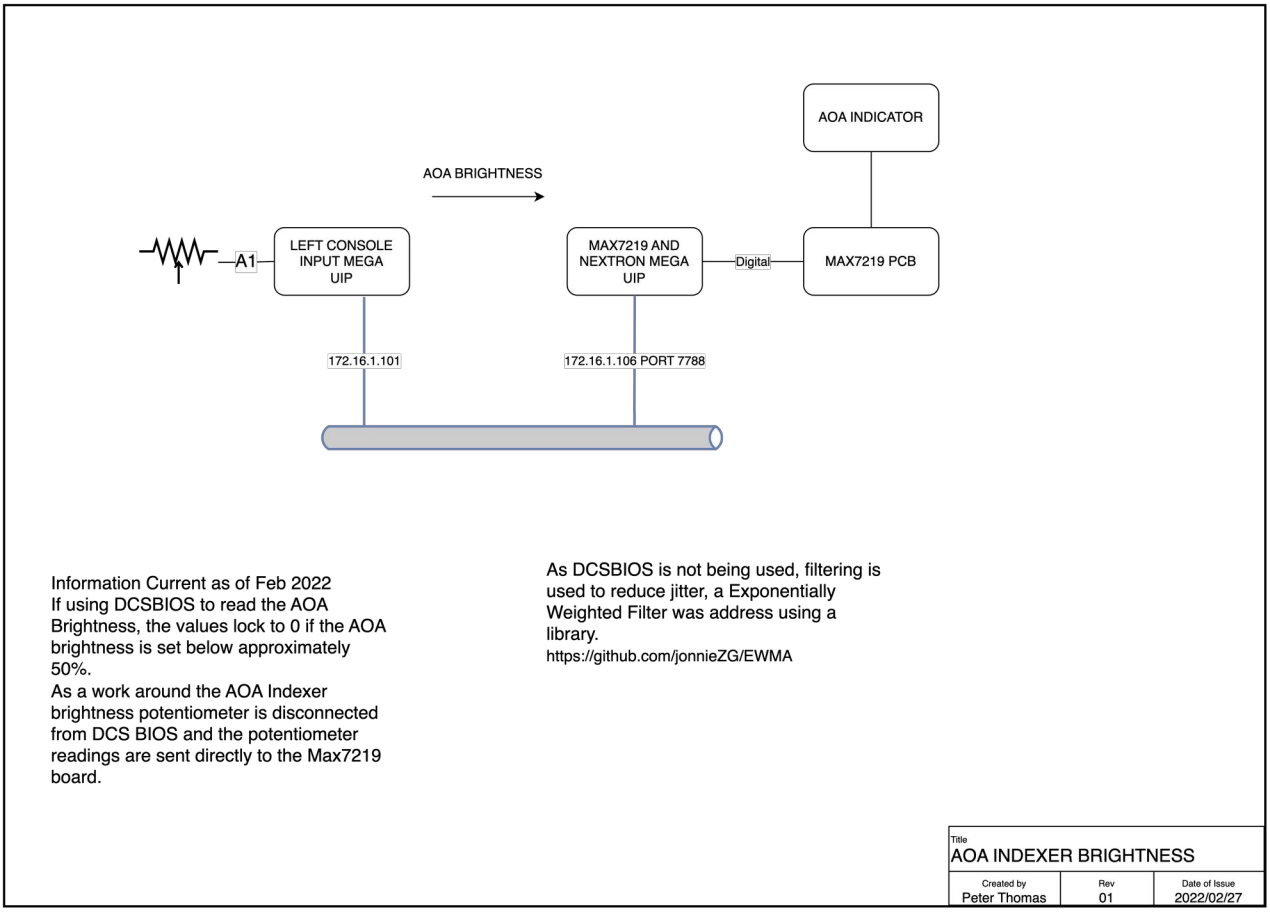
Right Console Altimeter

This was kindly put together by Ben.



UIP – AOA

The AOA Brightness is handled in a special way due to theDCS-BIOS behaviour at the time the pit was built. This did require a separate Max7219 to be used. A separate UDP port is used to listen for brightness updates on the Arduino servicing the Max7219s (Hornet Forward Output Upper Shield)



Software Modules

Preflight System Check

As the simulator is a complex entity, there are plenty of things to go wrong. A basic sanity test script checks connectivity of simulator components.

The script is located in ‘/Python HW Link/pyHWLink_System_Check.py’

It uses the pythonping module.

It pings the local loopback (127.0.0.1), its own IP Address (172.16.1.10), Arduino Ethernet cards, Rapsberry Pi the validates the correct wireless SSID (‘Shed’) is connected (obviously this is unique to my environment)

Computer Specifications

The following capabilities are desired

- Dual Wired Ethernet Interfaces, one for the Arduinos, the other for high performance connectivity such as last software updates and over the wire backups.
- WiFi interface for general connectivity
- Sound card with front and rear ports
- Six USB Ports
- Removable SSDs to facilitate high performance backups and swapping of simulator personalities (eg swap between DCS A10 and F16)
- Three high performance HDMI interfaces for the primary screens
- Ideally an HDMI port for the operators screens
- Four HDMI ports for the DDIs, AMPCD, and the standby ADI and RWR.
- An additional three video ports for the mirrors.

Computer Video Sub-system

With a minimum eight video ports there are some challenges in meeting these requirements. If mirrors are desired, a total of 11 ports are needed.

The original PC had a Nvidia 2080TI and a smaller three port Nvidia.

Ben found a four Port USB-3 video card which provides solid performance for DDIs and AMPCDs. The Startech USB32HD4 USB 3.0 to 4 HDMI Adapter - Quad Monitor lands around AU$270.

2023 Computer Specification

The PC was originally running a Nvidia 2080, it is intended to refresh the PC with a higher specification video card as a number of Blue Screens had occurred leading to the original PC not being able to boot.

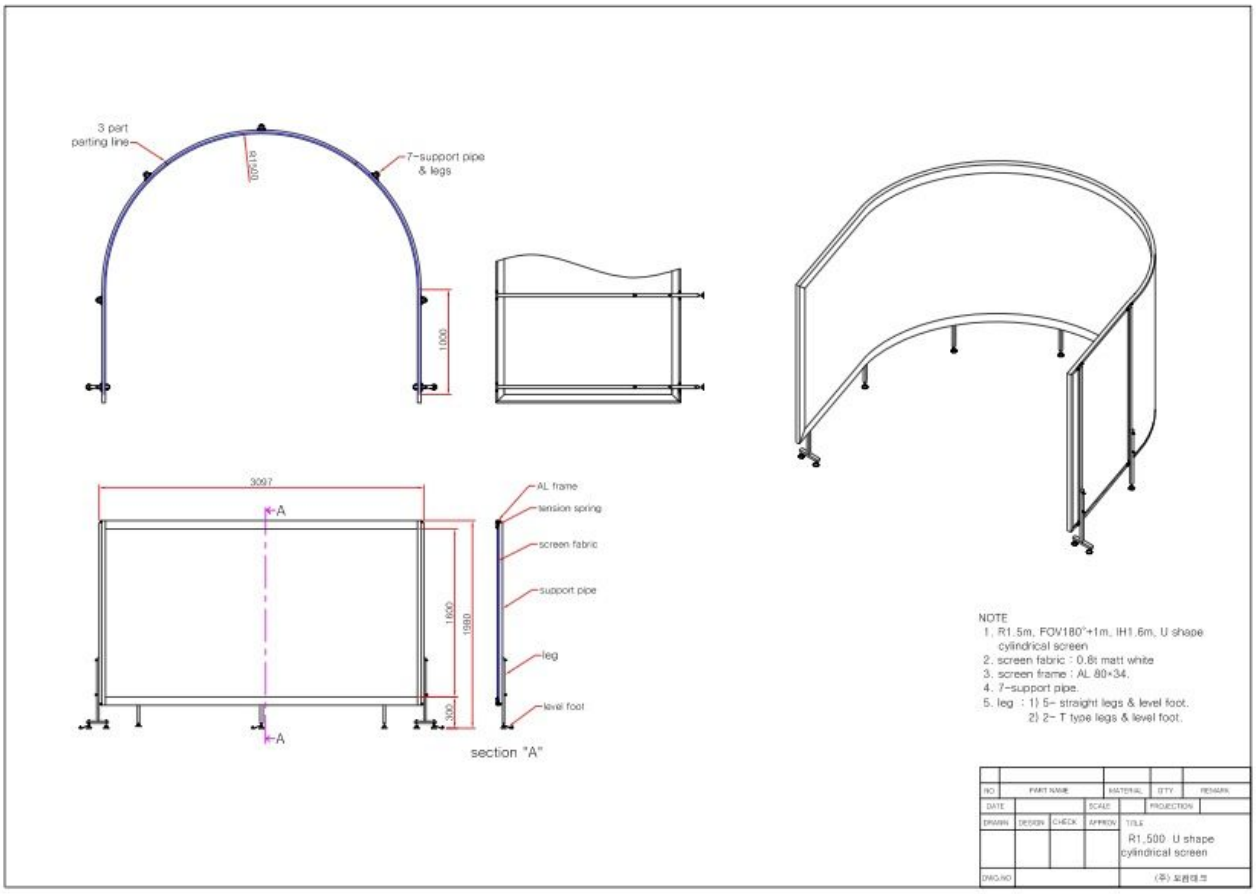
The replacement PC is this configuration.

- Intel S1700 Core i9 13900K 24 Core CPU BX8071513900K
- Corsair iCUE H150i RGB ELITE Liquid CPU Cooler CW-9060060-WW
- MSI S1700 E-ATX MEG Z790 ACE DDR5 Motherboard

- 64GB DDR5 (2x32G) Corsair 5200MHz CMT64GX5M2B5200C40 Dominator Platinum RGB RAM
- Samsung 990 PRO PCIe 4.0 NVMe M.2 SSD 2TB - MZ-V9P2T0BW
- Nvidia Quadro T600 4GB PCIe Video Card 900-5G172-2520-000
- Gigabyte RX7900XTX 24GB PCIe Video Card GV-R79XTX-24GC-B
- Corsair Full ATX Obsidian 1000D Tempered Glass Case Black (No PSU) PN CC- 9011148-WW
- 120mm Noctua NF-A12x25 FLX 2000RPM Fan
- 1200 Watt Thermaltake Toughpower GF3 Gen5 Power Supply PS-TPD-1200FNFAGA-4
- Microsoft Windows 11 Home 64bit DVD KW9-00632

The Curved Screen and Projectors

The curved screen was supplied by Mocomtech (https://mocomscreens.com). It has a radius of 1500m and a height of 1600mm. Once the design was complete using fly Elise Design Pro, the process of ordering and shipping was very smooth.



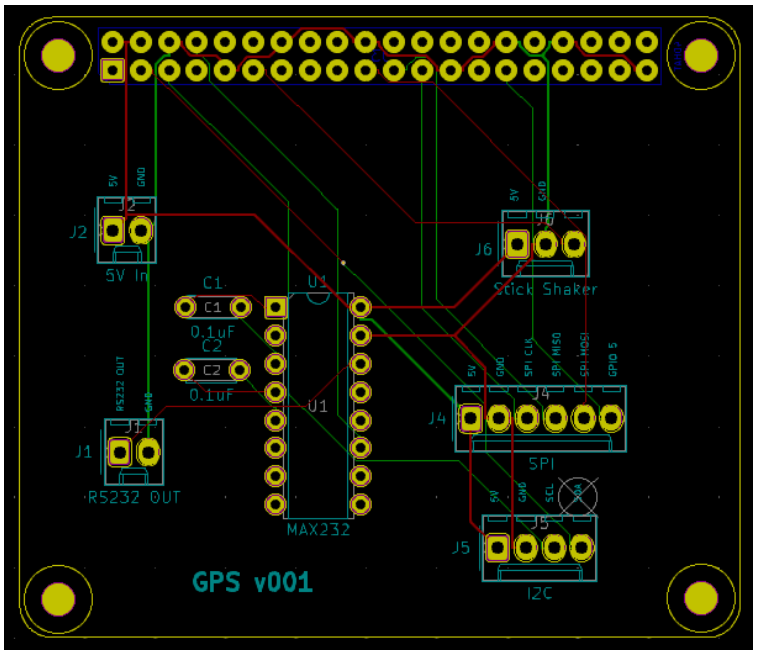
Three BENQ Short Throw 4K Projectors are used. With the high datarate associated with the 4K screens I found the older 10m HDMI leads no longer provided a reliable connection. I also have challenges with the Displayport to HDMI Adaptors.

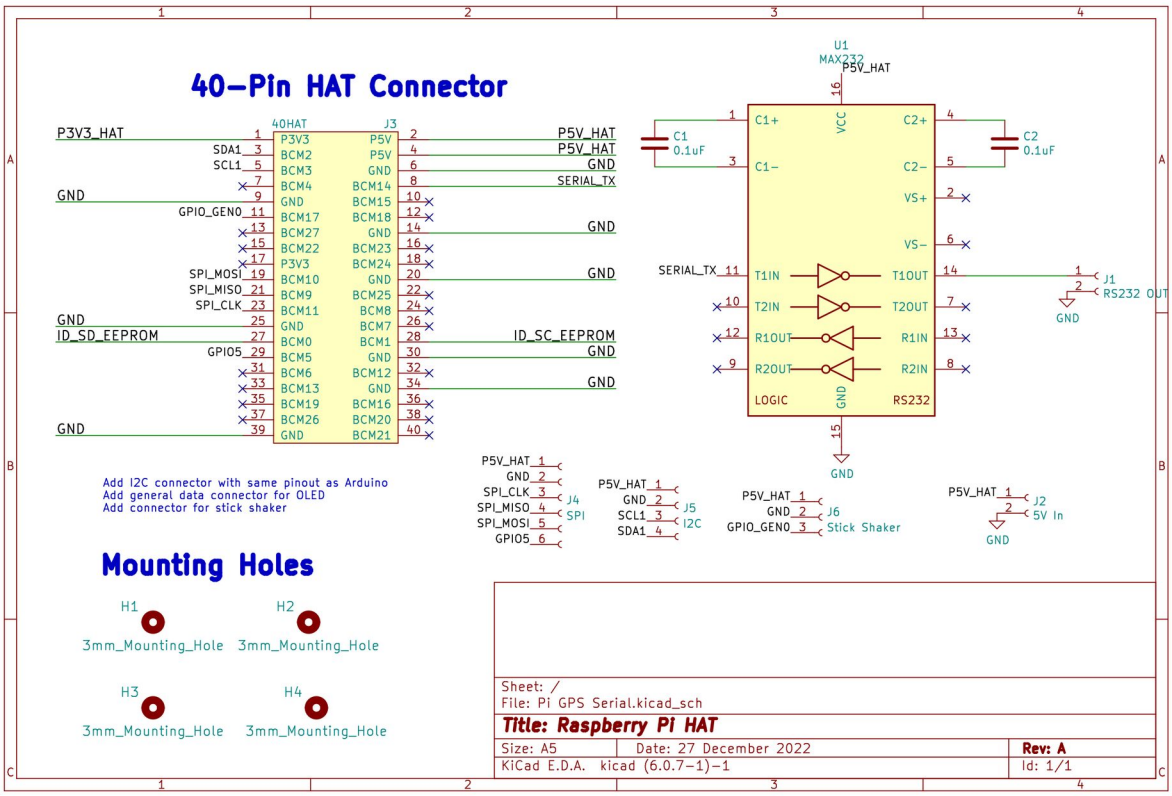
Printed Circuit Boards and Software

Left Console

Pi GPS Serial

This PCB converts the Raspberry Pi 3.3V Logic Levels to RS232 using a MAX232. It also has connectivity for SPI interfaces and a digital output for the 737 Stick Shaker.





UIP

Hornet Forward Output Upper Shield

This shield drives the Max7219 chain, PWM, Nextion, Compass, and Relay outputs. As of December 2022 it is version 5. Like many of the Mega shields in this project, the board is designed so it can be separated to support an Ethernet Shield. A 12V jumper is needed to connect the 12V rail between the two boards.

The PWM Dimmer drives flood lights, External lights and some of the gauge backlighting. The Mega only supports PWM on ports 2-13 and 44-46 (had to update the board as originally thought PWM was available on all Digital IO pins). MOSFETs drive the LEDs.

4 Ports drive relays which assist in providing a lights out experience for the pit.

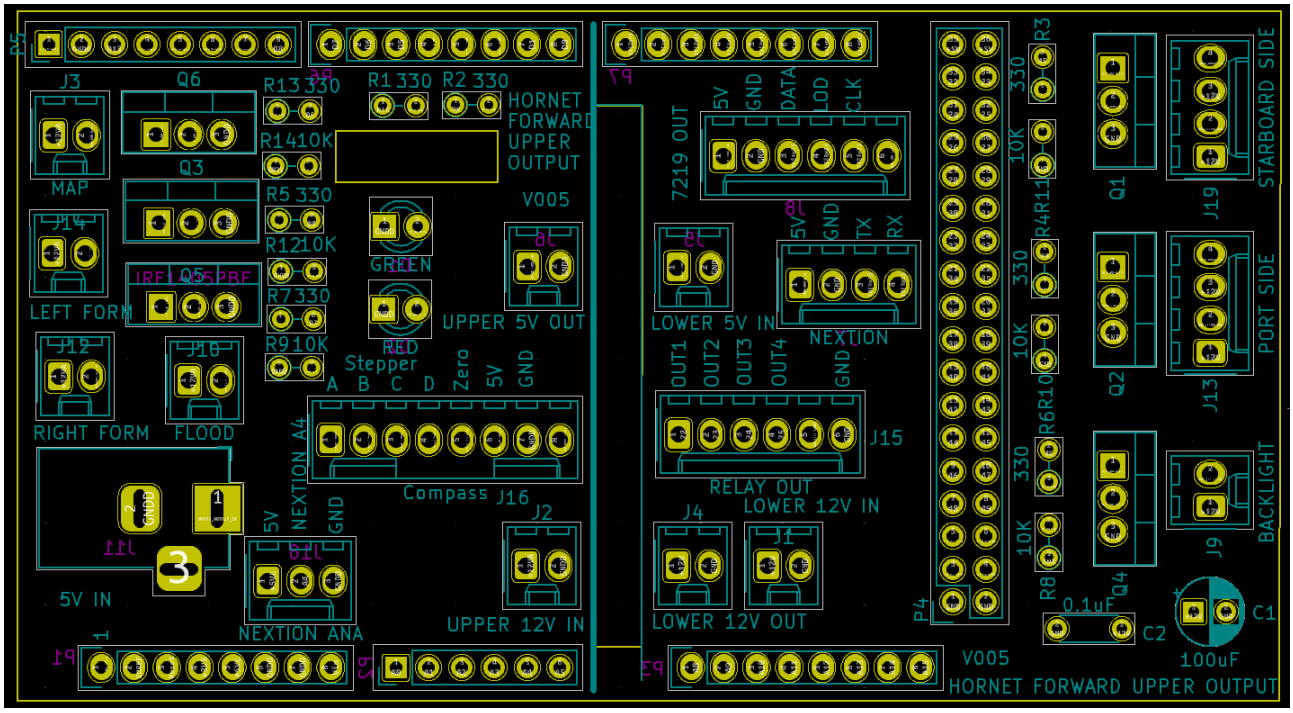
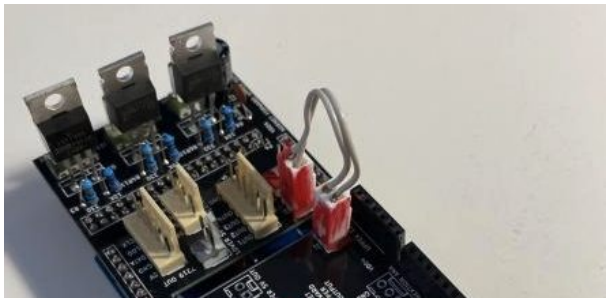


Figure 1: Hornet Forward Output Upper Shield - PCB Layout



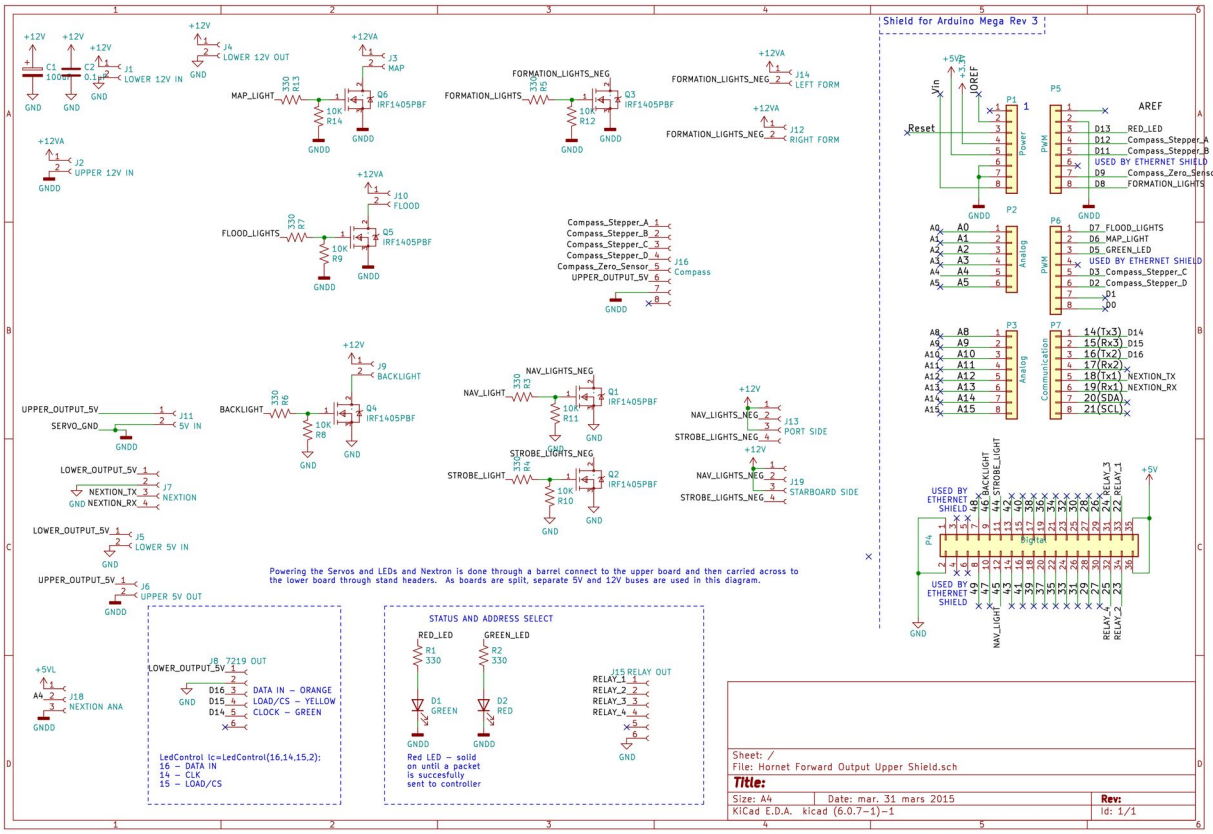
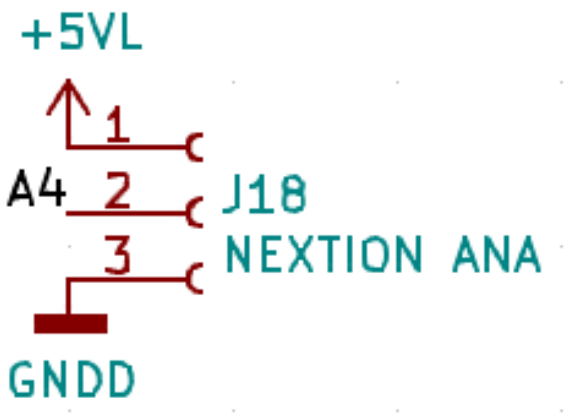
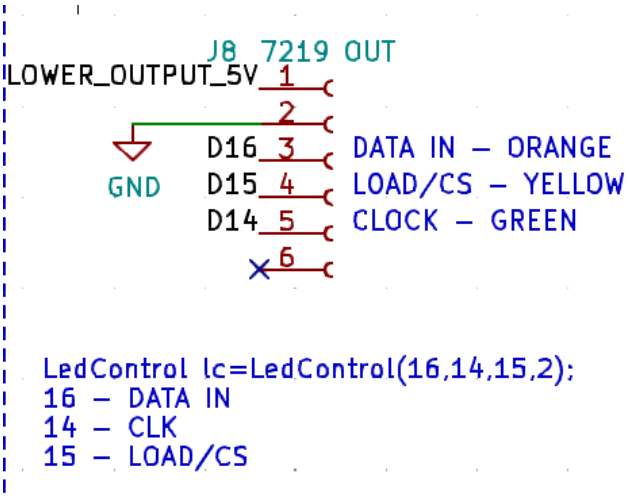
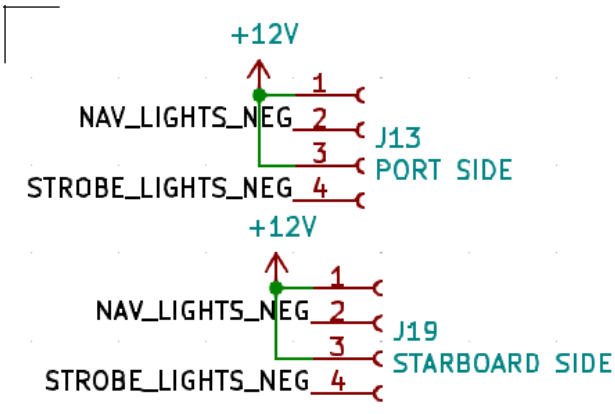
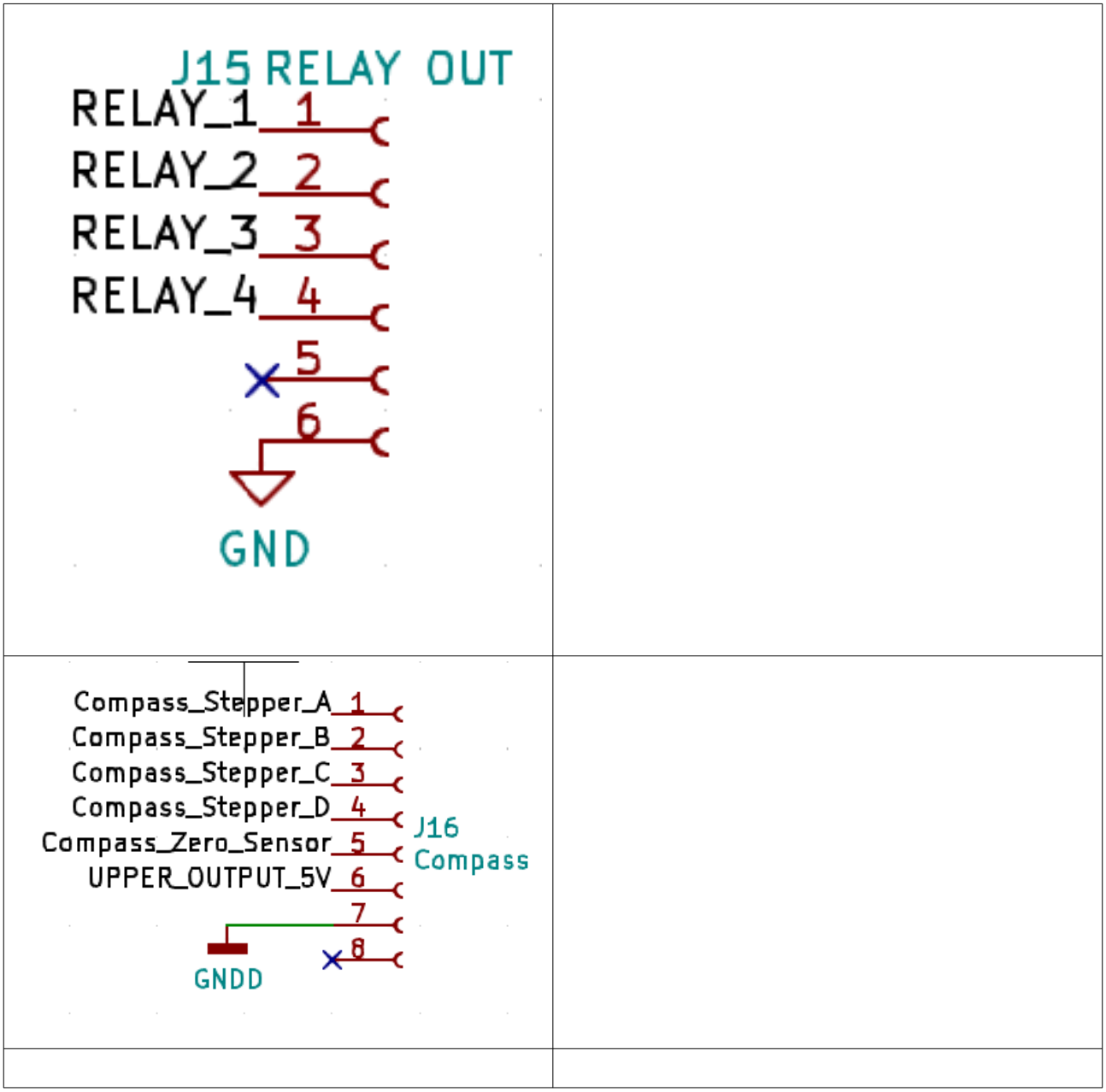


Figure 2: Hornet Forward Output Upper Shield - Circuit Diagram

	The +5VL is 5V lower. The Nextion is receive only for IFEI
	
	



Building the Computer

Building the Computer

Install Operating System

- Turn on visibility of hidden files and file extensions in explorer
- If using Samsung SSDs optionally install the Samsung Drive Monitoring Software
  - http://samsung.com/magician

Assign Static IP Address to Ethernet Connecting to Pit

Control Panel → Network and Internet → Network Connections. As the simple interface demands a default gateway and DNS be set, which is not used for this interface.

A static address (172.16.1.10/24) is used to receive switch events for MSFS 20202 as well as monitoring updates from the Arduino environment. If this address is not defined DCS interfaces will operate but updates may be slow due to the Arduino waiting for a timeout in resolving the ARP for the reflector. If it is not planned to use this link then is it recommended the following values be set to 0 in the Arduino code.

```
#define Reflector_In_Use 1
```

```
#define MSFS_In_Use 1
```

Install Documentation Tools

- These tools were chosen as they are cross-platform and lightweight
- LibreOffice - https://www.libreoffice.org/
- Draw.io – Offline installer – https://get.diagrams.net/.
  - Primary site - https://www.diagrams.net/

Install Development Environment

- Install Github –
  - http://desktop.github.com/
  - add primary repository -
    - https://github.com/bnepethomas/bne-arduino-flight-simulator-interfaces
- Instal Notepad++ https://notepad-plus-plus.org/downloads/
- Install Arduino IDE – including libraries used (built-in if author is not listed)

- https://www.arduino.cc/en/software
- FastLed 3.3.3 – Daniel Garcia
- Ethernet – 2
- Keyboard – 1.0.3
- Mouse – 1.0.1
- Servo 1.1.6 – Michael Margolis
- Stepper 1.1.3
- AccelStepper – 1.61.0 – Mike McCauley
- Adafruit GFX Library – 1.2.2
- Adafruit SSD1306 – 2.3.0
- LedControl – 1.0.6 – Eberhard Fahle
- Nextion – 2.0.2 – Helder Rodrigues
- U8g2 – 2.27.6 – oliver
- U8g2_for_Adafruit_GFX – 1.7.0 – oliver

- Install Python – minimum release 3.10 (for Case Statement support)
  - Add python.exe to PATH
  - Install python modules
    - pythonping (used in pyHWLink_System_Check.py)
    - SimConnect. Originally I had to use a fork on SimConnect, but the original project https://github.com/odwdinc/Python-SimConnect has now fixed the error, so using it.
      - `Pip install SimConnect (0.4.26)`
      - The module will be validated once MSFS 2000 has been installed.

Install Backup/Imaging Software

- Currently use Acronis True Image – but the perpetual licenses are no longer available – so may need to look for alternatives.

Install DCS

Install MSFS 2020

- Start MSFS 2020 and sign in
- Run ‘Python HW Link\pyHWLink_SimConnect_Transmitter.py’

Install Fly Elise

- Play the USB dongle in a port
- Install Fly Elise Display Pro (https://fly.elise-ng.net/immersive-display-pro/) Version 4.5.3
- Start Fly Elise Display Pro and Validate correct details are displayed in the license screen
- Install Fly Elise Calibration Pro Version 5.0.2. Download Zip, extract and then copy to a folder.
- Refer to “Building and Deploying A Curved Screen” for further details. Of note is the importance of allocating keystrokes to:
    - Start Warp
    - Stop Warp
    - Enable 2D Warping
    - Enable 3D Warping
- Start MSFS 2020 and validate screen is correctly displayed. MSFS SU 10 enabled Surround and multiscreen support which improves the curved screen experience. Largely it reduces the distortion encountered ad the edges of the screen, ideally meaning lampposts are no longer leaning over. The Fly Elise team have noted that the surround view support still has room for improvement (as of Oct 2022).
-

Install Orbx Central

- Note different pits may operate under different accounts
- Download scenery of interest

Configure DCS

- Start DCS and load desired modules (eg Syria, Hornet)

Install and Configure Flight Panels DCS-BIOS

- Connect the pits USB port
- Statically assign COM Port numbers for the Arduino’s using the noting the Arduino Serial Numbers
- Ben’s document provides a useful guide to setting up Flight Panels DCS-BIOS fork (DCS-BIOS-FLIGHTPANELS.pdf)
    - ***THE FOLLOWING STEPS NEED TO BE VALIDATED!***
    - Copy the SOCAT.EXE to the correct directory
    - Edit the batch file to reflect the serial ports in use.
    - Add a taskbar shortcut to the CMD file – Pin a CMD/Batch file to the Task Bar in Windows

Configure Keystroke Bindings for Magnetic Switches

Configure Monitors

- Now the Pit is installed we can complete configuration for the monitors and screens.
- Determine Monitor Layout. Optionally save the monitor layout using
- Configure Monitor related LUA files. Unfortunately DCS doesn't provide a simple way to configure multiple monitors.
- Configure RWR – can add rings https://forum.dcs.world/topic/241306-fun-with-rwr-exporting-and-a-question-on-scaling-a-texture-through-lua/

-

From Discord

Ben-111~~09~~/13/2021 10:10 PM

save to here

DCS\DCS World\Config\MonitorSetup

_ = function(p) return p; end;

name = _('4 Screen');

Description = 'F18-Pit'

Viewports =

{

    Center =

Expand

F18-C.lua

1 KB

add this to line 79 & 80 of file RWR_ALR67_init

in DCS\DCS World\Mods\aircraft\FA-18C\Cockpit\Scripts\TEWS\indicator

dofile(LockOn_Options.common_script_path.."ViewportHandling.lua")

- try_find_assigned_viewport("F18_RWR")

Applying DCS Updates

It is common for DCS to update the LUA files associated with the TEW and one other of the outputs. A reasonable approach is to keeping a renamed copy of the master in the same directory and then copying it over the updated files.

Simulator Startup

1. Power on the Projectors – wait for them to display an image
2. Power on the Simulator using the Power Switch located next to the throttle.
3. Monitor startup actions of gauges and back panel lighting
4. Power on the PC
5. run pyHWLink_System_Check.py to validate connectivity
6. Start SOCAT and validate all COM ports move to a good position.
7. Start Fly Elise (only 2d)
8. Start DCS
9. Start Flight
10. Change Fly Elise to 3d mode

Simulator Shutdown

1. Exit the Simulator
2. Shutdown SOCAT
3. Shutdown Fly Elise
4. Shutdown and fully power off the PC. This ensures nothing is receiving USB power via the PC
5. Shutdown the projectors
6. Power down the pit.

Building Tools

Tools used in the build include:

- Prusa MK3 filament printer
- Full Spectrum 30W CO_2 Laser
- Creality Resin Printer

To Do’s

1. I’m still yet to work out how you determine which build of DCS Flightpanels you are running on the PC. This is needed to determine if you are up to date as the library does receive active development.
2. Determine source of image drift on screens. Lock room temperature to 23 degrees for 30 minutes and monitor.

Components Used

Table of Contents

Backlighting

Connectors

	50sets Micro JST XH 2.54 2-8/10Pin 24AWG Single End 15cm Wire To Board Connector Cost $10 Note – the bulk of the connectors used in the project where JST, a small number of locations needed the Micro JSTs
	50Sets XH2.54 3Pin Single-Head Wire To Board Connector 15cm 24AWG With Socket Cost $20
	KF2510 2P-12P 2.54mm Straight/Right Angle Pin Header Housing Connector Terminals Cost $10

Display

	4K Type C to HDMI Adapter 30Hz USB C 3.1 Male to HDMI Female Cable FOR MACBOOK m Cost $11
	Ultra Flexible HDMI Cable 1M 10M 5M High Speed 1080p High Definition HD AU Stock Cost $17

External Connectors

	Keystone wall plate - 6 port white Cost $3 Supplier - homewired
	RCA keystone coupler - blue Cost $4 Supplier - homewired

	Cat6 RJ45 Keystone Inline Coupler * 2 Cost $7 Supplier - homewired
	USB 3.0 keystone coupler * 4 Cost $18 Supplier - homewired
	HDMI keystone coupler - 1.4 and 2.0 compatible Cost $10 Supplier - homewired
	Blank keystone inserts x 4 Cost $3 Supplier - homewired

Left Console – Comms Panel

	2pole 23step Rotary Switch Attenuator Guitar Selector Volume Potentiometer DIY Cost - $20 Note – Alternate pins from the upper and lower decks are needed to be used at this switch is a make before break
	LED Clearance Lights Side Marker Lamps White Amber Red Trailer Truck RV 12V-24V Cost $11 Used for external lamps

Miscellaneous

	25/100pcs Black Plastic Nylon M2 M3 M4 Hex Column Standoff Spacer Phillips Screw Cost $4

 	Matt Black Car Light Headlight Taillight Tint Vinyl Film Sticker Sheet Lamp Film Cost $11
Nuts and Bolts	Gateway bridge Brisbane https://www.boltandnut.com.au/
Possible knob supplier	electroenterprises MS21385-04CNN https://www.electroenterprises.com/
	https://www.digikey.com.au
	https://au.element14.com
	https://au.mouser.com

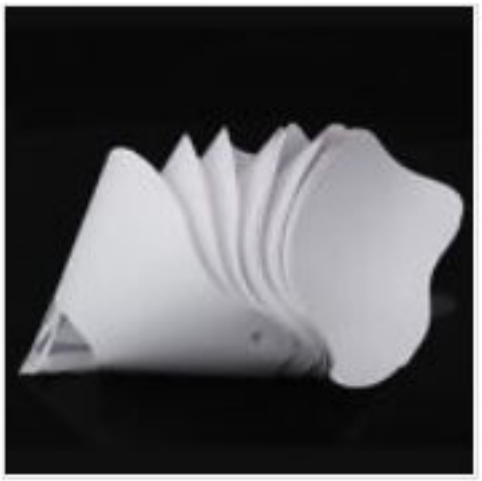
Mirrors

	7.9inch Capacitive Touch Screen LCD 400×1280 HDMI IPS Toughened Glass Cover Cost $105

LIP

	TF Micro SD to SD Card Extension Cable Extender Adapter SDHC For Car GPS TV 62cm Cost $11

Printing Support

	450Pcs Female Thread Knurled Nuts M2 M3 M4 Brass Threaded Insert Round Cost $25
	OZ 1080PC Stainless Steel Screws Bolts and Nuts Assortment M2/M3/M4/M5 Hex Head Cost $38
	Funnel Filter Paper 3D Printer Resin Disposable Photocuring Consumable Cost $13

Right Console

	10x T10 194 168 SMD 5050 LED Car Wedge Tail Side Parking Light Globe 12V – WHITE Cost $10 Used in forward flood light
	Light Socket for BA9S 1895 T4W 57 182 Globes and LEDs (Pack of 10) Cost $8 Used in most of the flood lights
	10x BA9S BAYONET LED LIGHT BULB 5SMD 5050 WHITE PARKER CAR GLOBE INTERIOR 12V Cost $9 Used in most of the flood lights
	LONELY BINARY N-Channel Power Mosfet IRF540 LOGIC LEVEL UNO R3 MEGA 2560 ESP32 Cost $25

Rudder Pedal Mounts

	**Ball Screw** BallScrew 1605 L250~2000mm SFU1605 End Machine &BK/BF12 &Nut Housing&Coupler Set Cost ~ $53
	Linear Rail Set 2X SBR16 300-2000MM Linear Rail Guide Slide Shaft Rod 4X SBR16UU Bearing Blocks Cost ~ $53

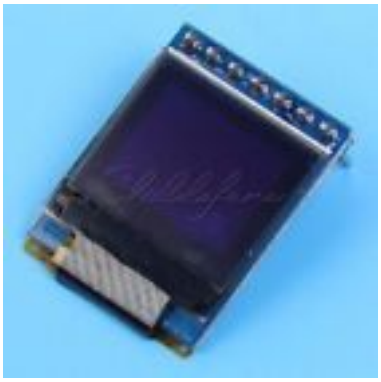
	DC12V-Metal Gear High Torque Turbine Turbo Worm Geared Reduction Motor 2-100RPM Cost ~ $19

Tools

	Doss Pro Coax & Lan Cable Tester Locates Distance to the Fault Coaxial Wire LCT8 Cost $209
	Brother PT-E300VP Industrial Labelling Machine W Li-ion Battery Heat Shrink Tape Cost $226
	Compatible with Brother Heat Shrink Tube HSE231 12mm Label Tape for P-Touch E300 Cost $14

	2in1 Soldering Solder Station Rework Iron Hot Air Gun Digital SMD Desoldering-AU Cost $90
	IWISS PH2.0 XH2.54 Dupont KF2510 JST Terminal Ratchet Crimper (SN28B + SN01BM) Cost $73

UFC

	0.91 inch IIC I2C SPI 128x32 White OLED LCD Display Module For Arduino PIC G3G8 Cost $6
	0.66 inch White 64x48 OLED Display Module SPI I2C 3.3-5V for Arduino AVR STM32 Cost $5

	10Pcs Micro Roller Lever Arm Open Close Limit Switch Kw12-3 PCB Microswitch Hot $3

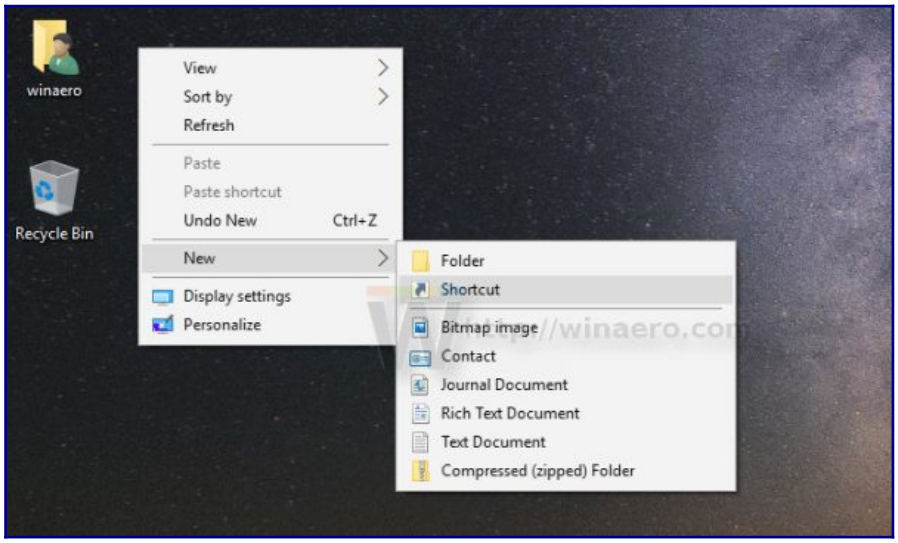
Tip and Tricks

Pin a CMD/Batch file to the Task Bar in Windows

Reference - https://winaero.com/pin-a-batch-file-to-the-start-menu-or-taskbar-in-windows-10/

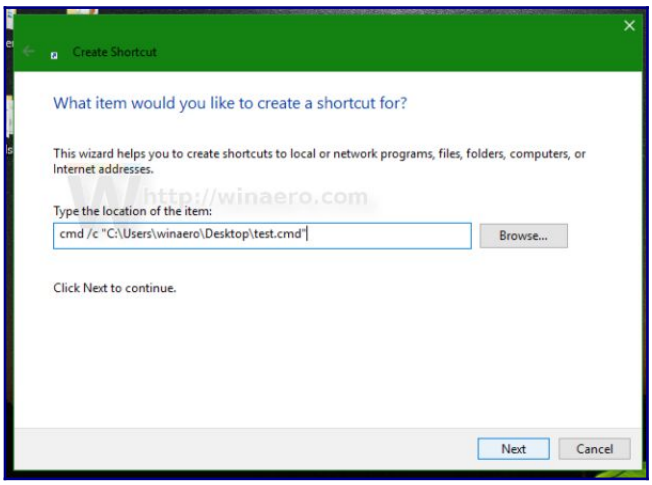
Here is what you need to do.

Right click on the Desktop and select the command "New -> Shortcut" from the context menu.

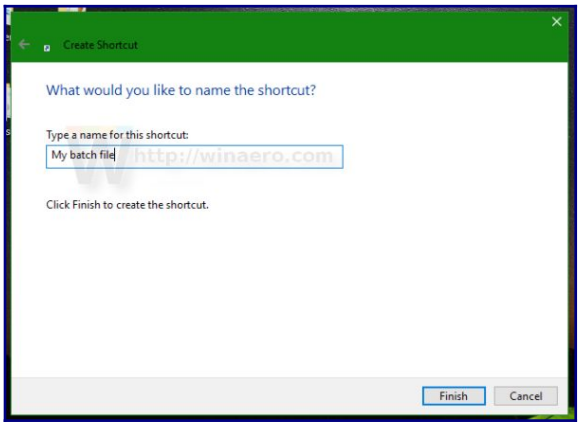


In the shortcut target, type the following text:

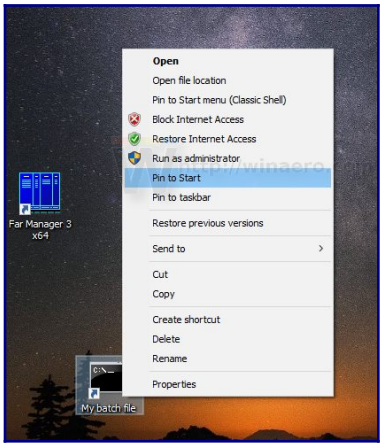
```
cmd /c "full path to your batch file"
```



1. Name the shortcut and change its icon if you need:



2. Now, right click the shortcut and choose "Pin to Start" from the context menu:



Useful URLS

https://www.openflightschool.de/

Lots of tips and tricks

https://www.openflightschool.de/mod/book/view.php?id=236

https://forum.dcs.world/topic/134049-3-monitor-setup-moving-all-radio-texts-to-center-screen/

Adjusting ATC menu position

https://forum.dcs.world/topic/258724-how-to-multi-monitor-mfcd-display-export-set-up-guide-nov-2022-updated/

http://breakthrusoftware.com/html/onlinedocs/kb/installkb/ScreenCoordTool.html

Possibly useful tool for screen coordinates

https://sourceforge.net/projects/monitorswitcher/

Monitor Configuration Save and Switcher – may be useful when monitors are being connected and disconnected.

https://winaero.com/pin-a-batch-file-to-the-start-menu-or-taskbar-in-windows-10/

Pin a CMD/Batchfile to the Taskbar

Plane Spotting

https://aircraftnoisemap.airforce.gov.au/assets/site.html?384#base/5/region/77/events

http://aussieadsb.com/airspaces?fbclid=IwAR3qWMav7fapFhkGdy8n6r5MlbFH38-_HqVpLdtuZBgai6TSzfANd0g0efg

Table of Figures