The Jet Ranger Build

Todo’s 2

The Build Process 2

CKAS Platform 2

Output Card 3

IP Port Utilisation 5

GPS Out 5

GPS Setup 5

Driving the OLEDs 7

Initialisation Sequence 7

Transmit Data to oled 7

Native LUA data 8

Preformatted data from bne-ocr-oled 9

The Arduino Sketch 9

Overhead Panel 12

Configuring Switches 13

P3D Installation 15

Sound System 27

FSX Data flows for radios and lights 32

Data Flows 33

Data Flows with Protocol and Ports 34

Devices 35

I2C Demux Schematic 36

# Todo’s

* Update Rotaries to stacked rotaries from Leo Bodnar
* Determine number of rotaries to set priorities
* ADF needs a 3rd rotary, in the mean time map the RH rotary, clockwise to increment 0.1’s, counter clockwise to increment 1s. LH rotary, clockwise to increment 10s, counter clockwise to increment 100s.
* Valid VOR tones are coming through headsets.
* Sent Grip to Jess
* Get to why Dodo isn’t playing with P3D 2.3
* Determine how to back up Linda configurations for Joysticks.
* Update table of output ports for SIOC USB Outputs card.
* Draw out audio system structure
* Remove fan interference when they are switched, either through the addition of ceramic caps, or using separate PSU for fans. This will mean modifying the overhead, panel so perhaps first test using plug pack PSU on a single fan, noise could be introduced through the common ground, so it may be easier to separate out the 5V supply and move it to the base. Other option is to use a 9V battery, would ideally use a rechargeable.
* Used a separate 5V supply and noise disappeared.
* Remove hum from overhead speakers, suggest grounding both (Left and right) inputs to amp to ground. Should be able to do this as rotary switch has 4 poles.
* Look at using DLL to get Dodo values to drive leads, code fragment is included dodo hardware interface document.

# The Build Process

FSX installed

Dodo, ran up no gauges displayed

Installed FSX Acceleration, Dodo gauges now displaying correctly

Installed DoDo SP2

Reasked about trusting the Dodo plugin

Installed FSUIPV

Pasted panel.cfg into the low skid

Panel undocked

On the Russ Cool Panel, the difficulty setting knob is located underneath the Temperate Display.

# CKAS Platform

Initially had an older CKAS driver, which exploded when running P3D.

The Drivers can be found at [www.ckas.com.au/downloads/Setup%20CKAS%20SimCor.exe](https://mail.cisco.com/owa/redir.aspx?C=bXwYACrCLk60DR1P-Hgiu6Hw94VFudEIiODg9oW5F09uz8ZOVHTiw1gova1JRNpxLZvw7KfOxqg.&URL=http%3a%2f%2fwww.ckas.com.au%2fdownloads%2fSetup%2520CKAS%2520SimCor.exe" \t "_blank)

As of Oct 2014 the file version is 1.0.0.21

# Output Card

Open Cockpits (OC) hardware is both cost effective, and once you’ve got your head around it, very flexible, the biggest challenge is getting started.

To test the USB output card use test\_outputs.exe, this is a simple to use GUI for turning outputs on/off. It can also be used to see what are the device numbers associated with each card. Note these device numbers are likely to change if you inserted them into a different USB port. If you are unsure which is which ground one of the analog inputs, the other inputs will jump around.

*For testing LHS is 169, RHS is 114*.

The modules generally communicate to the hardware via a SOIC Server that uses compiled scripts to map incoming variables from the simulator to OC hardware. The output module needs to be linked to script variables.

1. Start SOIC Server
2. If the OC card is not already plugged in, plug it in
3. You should see an entry in the devices box in the top right hand corner
4. Click the config devices button
5. If the device is not already configured you’ll see the card’s ‘Script Device IDX’ is not defined, and the Master Status is ‘not linked’ in the right hand window. To link, click the USBOutputs line on the left hand box, and the Master entry in the right hand box, then the link button.
6. Save and Exit.

*For testing Mapped 169 as Master (IDX=0) and 114 as IDX = 1*

If you are using multiple USB Output cards, one will be the Master, the second will be unlink. Click on the card whose IDX is ‘Not Defined’ and enter a script IDX in the center text box. Click ok and save.

*For testing – normally do not specify Device Number (Dispositivo?), assigned Device Number 1 to T/O Trim 191. Done by click Edit Script, Double Clicking on Var and assigning value. File Save, Close Window, click Reload.*

*Now go into IOCP Console, Connect, Log On. Set value of Var 191 to 1. Test switch holds position.*

*For final test, added another card, Device 164 in USB Outputs. Changed Device IDs, 169=0, 164=1, 114=2.*

Dimming USB Outputs

As two of the three USB Output cards are driving leds, they should be dimmable. The card driving Magnetic switches and relays should be a fixed output.

Here’s the sample script:

Var 0002, name dimmer, Link IOCARD\_ANALOGIC, De

vice 1, Input 1, PosL 1, PosC 128, PosR 255

// potentiometer reading

{

&bright = &dimmer

}

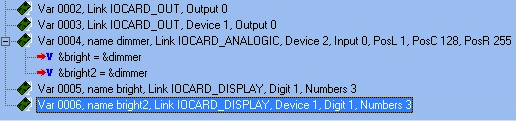
Var 0003, name bright, Link IOCARD\_DISPLAY, Device 1, Digit 1, Numbers 3 // dimmer at USB Outputs

The steps to create the script:

Go into Script Editor and create all the variables (basically assigning a name to an output or handle).

Once this is done click on the input value that will be used to set the outputs, and choose edit-> New Command. From there you create an assign of source var to destination variable. In the GUI the commands should appear under the input variable. Note – you need to ensure you have selected the input variable when adding the commands

*Var 0003, Link IOCARD\_OUT, Device 1, Output 0  
  
Var 0004, name dimmer, Link IOCARD\_ANALOGIC, Device 2, Input 0, PosL 1, PosC 128, PosR 255  
{  
  &bright = &dimmer      
  &bright2 = &dimmer      
}  
  
Var 0005, name bright, Link IOCARD\_DISPLAY, Digit 1, Numbers 3  
  
Var 0006, name bright2, Link IOCARD\_DISPLAY, Device 1, Digit 1, Numbers 3*



The Output Cards variables are mapped in the primary .ssi associated with the SOIC server.

# IP Port Utilisation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Application | Client/Server | TCP/UDP | Port | Partner Applications |
|  |  |  |  |  |
| SOIC Server | Server | TCP | 8092 | IO Card LCD, |
| IO Card LCD | Client | TCP | 8092 | SOIC Server |
| SOIC\_Conv | Client | TCP | 8092 | SOIC Server |
| SOIC\_Conv | Server | UDP | 7777 | DCS World – Receives comma delimited data from DCS |
| SOIC\_Conv | Server? | UDP | 7779 | DCS World? |
| SOIC\_Conv | Client | UDP | 7780 | DCS World  CDU Keystrokes send to DCS |
| DCS World | Server | UDP | 7780 | SOIC\_Conv – Receives CDU Keystrokes from SOIC\_Conv |
| Arduino Display | Server | UDP | 7781 |  |
| GPS Out | Server | UDP | 7782 | Receives UDP packets from BeforeNextFrame and reformats to serial with checksum for GPS |
| Simmeters | Server | UDP | 6061 | DCS World – Receives comma delimited data from DCS |

# GPS Out

Use native interface in FSUIPC. Note after you have added the GPS configuration in FSUIPC, it does not send anything out to the GPS until you restart FSX. It also identified that COM7 for the Arduino was in use, so didn’t offer it up as a choice.

# GPS Setup

The GPS takes a serial NEMA data feed.

System Setup

Communications Port

Baud Rate 19200

NEMA Input

NEMA GPS Input

Configure NEMA

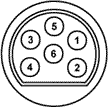
GLL, RMC and RMB, GGA, GSA and GSV, APB

Software version 1.7.0

Cable Pinout

DB9

|  |  |  |  |
| --- | --- | --- | --- |
| DB9 |  |  | Lowrance |
| 1 | Gnd | Black | 3 |
| 2 |  |  |  |
| 3 | RX RS-232 | Green | 2 |
| N/C |  | White | N/C |
| N/C | +12V | Red | 1 |

[](http://www.allpinouts.org/index.php/File:Conn_lowrance_002.gif)  
Pin connector as seen from back of Airmap 2000C

|  |  |  |
| --- | --- | --- |
| **Pin** | **Name** | **Description** |
| 1 | VCC | Power |
| 2 | RX | RS-232 (receive serial) |
| 3 | GND | Ground |
| 4 | TX | NMEA/Pseudo RS-232 (transmit serial) |
| 5 | - | DO NOT USE (recharging purposes) |
| 6 | - | DO NOT USE (recharging purposes) |

Was unable to buy an original NEMA cable, so had to modify a power only cable.

Modifying a power cable to a NEMA cable

1. Purchased a DB25 headshell with pins to be placed.
2. Drill out pin 2 of existing existing power connector with a 1mm drill all of the way through to the back for the cable.
3. Drill out pin 2 with a 3mm drill only partially so that the pin is able to seat in the socket without being able to be pushed all the way through
4. Heat Glue DB25 socket connector in Lowrance connector.

# Driving the OLEDs

The OLEDs use an I2C bus, which enables all of the OLED displays to be connected to a single Arduino card. The only challenge encountered when using this approach is that a ‘demultiplexor’ (demux) is needed as all of the OLED displays have the same I2C bus address.

In addition to enabling the OLED displays to be uniquely addressed in S/W. the demux buffers the output which reduces the likelihood of issues associated with longer cable runs. Uniqueness is achieved by ANDing SCL with one of the Digital outputs. Initially SDA was also going to be buffered, but for reasons I’m yet to work out, the displays did not appear to receive data when SDA was buffered, if may be due to clock-skew and issues associated with not having pull-up resistors. As the system is currently working, I’m reluctant to attempt to fix it. The displays are never read, only ever written to.

Another benefit of using common OLEDs is initialisation codes can be ‘broadcast’ to all attached displays, this is done by raising all of the Data Outputs to high. This is used to initialise all of the attached displays.

Generally ‘clearing’ displays is avoided as this incurs a performance hit. Displays are updated an entire line at a time, with trailing spaces to pad out the display.

As the Arduino is presented via a USB-Serial interface there is a ‘serialisation tax’ which must be taken into consideration when sending data to the Arduino card. The performance tax is minimised by setting the UART to run at 115,200bps. As a very simple protocol is used to send data to the displays, the rate at which updates are send must not cause ‘tail drop’ in the UART, this will result in corrupted data being displayed. A 50mS delay appears to be adequate during most updates. A good test of timing is to do a chaff/flare dump.

Protocol between Arduino-multi-oled-receiver and Arduino card:

## Initialisation Sequence

Write out 2 \* Nulls - \_serialPort.Write(NULL, 0, 1);

Clear each display - ClearArduinoDisplay(oled1)

Send information text to each display - SendToArduino(oled1, FirstLine, "1 Waiting for DCS");

## Transmit Data to oled

*SendToArduino(oled6, FirstLine, lastVHF\_FM.PadRight(16));*

|  |  |
| --- | --- |
| Parameter  No |  |
| 1 | Oled Number – oled 1 to oled 6 |
| 2 | Line Number – either FirstLine or Second Line |
| 3 | Text to be displayed – ASCII text – should be padded to end of display |

There are two routines that actually send data to the Arduino

static void SendToArduino(byte[] oledPtr, byte[] lineNumber, string StrToSend)

{

if (\_serialPort.IsOpen)

{

\_serialPort.Write(STX, 0, 1);

\_serialPort.Write(oledPtr, 0, 1);

\_serialPort.Write(lineNumber, 0, 1);

\_serialPort.Write(StrToSend);

\_serialPort.Write(ETX, 0, 1);

}

}

static void ClearArduinoDisplay(byte[] oledPtr)

{

if (\_serialPort.IsOpen)

{

\_serialPort.Write(STX, 0, 1);

\_serialPort.Write(oledPtr, 0, 1);

\_serialPort.Write(FirstLine, 0, 1);

\_serialPort.Write(CLS,0,1);

\_serialPort.Write(ETX, 0, 1);

}

}

This is called from branches which do a comparison against the last line sent to a given line on a given display.

There are two sources of data for the OLED displays, the first is data directory from DCS via LUA script, the second is from the ‘memory extractor’, cryptically called ‘bne-ocr-oled’, as this initially used Overpro’s OCR code, but now uses Gravils? Memory extractor. Both of these deliver the data over UDP using a variable name = value string. The data from the memory extractor is identified with a ‘9999’

// CMSC/CMSP has a tag attrbitue of 9999

if (Get\_Value(words,9999) == " ")

The bne-ocr-oled generated the entire line to be displayed, whereas the data delivered using the native LUA interface needs to be formatted.

### Native LUA data

*// VHF*

*sWorkstr = "VHF FM " + Get\_Value(words, 143) + Get\_Value(words, 144) + Get\_Value(words, 145) + "." + Get\_Value(words, 146);*

*if (sWorkstr != lastVHF\_FM)*

*{*

*lastVHF\_FM = sWorkstr;*

*SendToArduino(oled6, FirstLine, lastVHF\_FM.PadRight(16));*

*Thread.Sleep(20);*

*}*

### Preformatted data from bne-ocr-oled

if (returnData.IndexOf("CMSP") > 0 )

{

if (returnData != lastCMSP)

{

SendToArduino(oled7, FirstLine, returnData.Substring(12, 18));

Thread.Sleep(50);

SendToArduino(oled7, SecondLine, returnData.Substring(32));

Thread.Sleep(50);

lastCMSP = returnData;

}

}

## The Arduino Sketch

Unlike a UDP packet, an asynchronous serial port does not intrinsically have a Start-Of-Message, End-Of-Message construct; this has to be defined using a higher-level protocol. The Arduino runs a Finite State Machine to keep track of state.

*// FSM States*

*#define stateWaitingForSerialData 0*

*#define stateLookingForStartOfMsg 1*

*#define stateLookingForOLEDPtr 2*

*#define stateLookingForLinePtr 3*

*#define stateRxAndDisplayChar 4*

*#define minoledPtr 1*

*#define maxoledPtr 7*

*#define minLinePtr 1*

*#define maxLinePtr 2*

*// Map Output Pins to OLED Displays*

*int oled1 = 2; // First usable output pin is D2 so all pins are incremented by 2*

*int oled2 = 3;*

*int oled3 = 4;*

*int oled4 = 5;*

*int oled5 = 6;*

*int oled6 = 7;*

*int oled7 = 8;*

After the hardware initialization strings have been broadcast to all attached OLEDs, the number of each OLED is displayed, this is primary for troubleshooting, there is a brief delay, and then waiting for DCS is send to al displays.

In addition to the OLEDs themselves, each of the AND gates selection legs has an LED which is illuminated whenever that gate is selected.

Arduino Sender

A simple interface that drives multiple OLED displays off a single Arduino. The formatting of strings is performed on the host PC, leaving the Arduino to act as a Serial to I2C bridge/mux. As the OLED displays have the same target I2C address on the I2C bus the system must either have a bunch of Arduino’s (which means more USB ports and space consumed in the pit), or a muxing mechanism is added to software select the display to send data.

Principles:

Keep the Arduino code real simple, basically it receives a 5 part message

1. SOM (Start of Message)

2. Display Ptr (which display is the text going to)

3. Line Ptr (the original displays had two lines, so select which line the text is destined for)

4. Text (including trailing spaces to overwrite all previous displayed text)

5. EOM

The I2C bus is unidirectional ,i.e. the displays are never queried, it is assumed the data was received.

Start-up

On start the Arduino initialises all OLED displays, raising all Output

Hardware:

1. Currently a 4081 Quad AND gate is used. For some reason the I2C Clock (A4 pin) cannot be buffered via a AND gate. So there may be issues with Bus loadings as displays are added. It also could be due to not having a pull up resistor on the input to the gate

2. The initial design had the LEDs tied to the +5V rail, this means they are lit when the output is low.

3. Output pins D0 and D1 are used for internal purposes, so the first pin used is D2.

|  |  |  |
| --- | --- | --- |
| Variable | O/C Variable | OC OutputPort[[1]](#footnote-1) |
| Com1 Transmitt | 100 | 48 |
| Com2 Transmitt | 101 | 50 |
| COM RxBoth | 102 | 49 |
| Nav1 Sound | 103 | 44 |
| Nav2 Sound | 104 | 43 |
| Marker Sound | 105 | 42 |
| DME Sound | 106 | 41 |
| ADF1 Sound | 107 | 40 |
| Inner Marker | 110 | 45 |
| Middle Marker | 111 | 46 |
| Outer Marker | 112 | 47 |
| Eng Out | 120 | 23 |
| Rotor Low | 121 | 24 |
| Fuel Low | 122 | 33 |
| Generator Fail | 123 | 38 |
| Battery Warm | 124 | 31 |
| Battery Hot | 125 | 32 |
| Fuel Pump Fail | 126 | 39 |
| Transmission Pressure Fail | 127 | 25 |
| Transmission Temperature Fail | 128 | 26 |
| Engine Chip | 129 | 30 |
| Tail Rotor Chip | 130 | 21 |
| Transmission Chip | 131 | 29 |
| Fuel Filter Fail | 132 | 22 |
| Baggage Door | 133 | 20 |
| SimConnect Fail | 134 | 28 |
| Spare1 | 135 | 37 |
| Spare2 | 136 | 36 |
| Spare3 | 137 | 35 |
| Spare4 | 138 | 34 |
| Spare5 | 139 | 27 |
| AP\_Master Switch | 300 |  |
| AP\_Nav1 Lock | 301 |  |
| AP\_Heading Lock | 302 |  |
| AP\_Alt Lock | 303 |  |
| AP\_IAS Lock | 304 |  |
| AP\_Appr Lock | 305 |  |
| AP\_BC Lock (Back Course) | 306 |  |
| AP\_FD | 307 |  |

# Overhead Panel

The Map reading light has it’s own switch, which is not mechanical, this means it cannot be controlled via an external toggle.

From back 1 is left

Power Distribution (need to add 12V permanent output for flexi light)

1: 12V Input

2: GPS

3: Cockpit Lights

4: Not Connected

5: 5V Out to intercom

|  |  |  |
| --- | --- | --- |
|  | Pins |  |
| HTR | 7 – 11 |  |
| Dir Gyro | 29 |  |
| Pitot Head | 61 |  |
| Avionics | 45 |  |
| Battery | 60 |  |
| Start-Gen | 44 |  |
| Heat-Vent - Heat | 13 |  |
| Heat-Vent – Vent | 12 |  |
| Gen Field | 28 |  |
| Gen Reset | 59 |  |
| Anti Coll | 27 |  |
| Pos Light | 43 |  |
| Fuel Boost Aft | 26 |  |
| Fuel Boost Fwd | 42 |  |
| Caution Light | 58 |  |
| Landing Light | 25 |  |
| Inst Light | 41 |  |
| Cockpit Light | Power Dist | Panel Backlighting |
| Strobe | 57 |  |
| Aud | Power Dist | Intercom |
| GPS | Power Dist |  |
| Co-Pilot Blower | Power Dist |  |
| Pilot Blower | Power Dist |  |
| Inst Light Dimmer | 3 Pin to Power Dist |  |
|  |  |  |
|  |  |  |
|  |  |  |

Inputs

|  |  |  |
| --- | --- | --- |
| Jaycar Keypad | | |
| 3+2 | 1+2 | 5+2 |
| 1 | 2 | 3 |
| 3+7 | 1+7 | 5+7 |
| 4 | 5 | 6 |
| 3+6 | 1+6 | 5+6 |
| 7 | 8 | 9 |
| 3+4 | 1+4 | 5+4 |
| \* | 0 | # |

## Configuring Switches

High Level Steps

Clear Axis for all joysticks

For Dodo need to specifically map throttle cutoff

Clear switch mapping for both toggles and rotaries

Suggest using first value at the top of the list when configuring. Click to remap action, if the switch is available/unused, then move to next switch, and then clear.

Use list of overhead switches to verify which ones can be mapped. One that may not have been done if the warning circuit breaker.

Preference would be to use the native interface in FSUIPC, as hopefully it is more timely than LINDA. There are limitations however, FSUIPC only recognises the first 32 buttons, so perhaps LINDA is more appropriate.

Suggested panel order

Radio Stack – not forgetting the Transfer/Switch button

ADF Panel

The DodoSim 206 FSX does not require FSUIPC to operate.  All the switch functions can be mapped using standard and non-standard key assignments as detailed within the manual.  However, most of these are toggle functions and therefore not ideal for use in hardware cockpits where specific On/Off or Fwd/Both/Off (as in landing lights) switch commands are needed.  
  
Recent versions of FSUIPC4 for FSX offers a useful feature which the 206 (v1.054) makes use of, giving pit builders a whole host of new potential event bindings.  Here's how to use them:  
  
1) Create a text file in notepad within the 'Flight Simulator X/Modules/' folder alongside the FSUIPC4.dll called 'DodoSim206FSX.evt' (The file folder is called Events  
The entry inside the folder is  EventFiles)  
  
2) Put the following text within the file:

|  |
| --- |
| **Code Sample** |
| [Events] 1=DODOSIM206FSX.ELECT\_FUEL\_PUMP1\_SET 2=DODOSIM206FSX.ELECT\_FUEL\_PUMP2\_SET 3=DODOSIM206FSX.WARNING\_HORN\_CB\_SET 4=DODOSIM206FSX.NAV\_LIGHTS\_SET 5=DODOSIM206FSX.DIFFICULTY\_LEVEL\_SET 6=DODOSIM206FSX.DIFFICULTY\_LEVEL\_INC 7=DODOSIM206FSX.DIFFICULTY\_LEVEL\_DEC 8=DODOSIM206FSX.HEAT\_VENT\_SET 9=DODOSIM206FSX.GENERATOR\_FIELD\_CB\_SET 10=DODOSIM206FSX.GENERATOR\_RESET\_CB\_SET 11=DODOSIM206FSX.DIR\_GYRO\_SET 12=DODOSIM206FSX.MASTER\_BATTERY\_SET 13=DODOSIM206FSX.GENERATOR\_SWITCH\_SET 14=DODOSIM206FSX.ANNUNCIATOR\_TEST\_SET 15=DODOSIM206FSX.WARNING\_MUTE\_SET 16=DODOSIM206FSX.TURBINE\_OVER\_TEMP\_TEST\_SET 17=DODOSIM206FSX.GPS\_DRIVES\_NAV1\_SET 18=DODOSIM206FSX.FUEL\_VALVE\_SET 19=DODOSIM206FSX.CAUTION\_LIGHT\_SWITCH\_SET 20=DODOSIM206FSX.ANTI\_ICE\_SWITCH\_SET 21=DODOSIM206FSX.HYDRAULICS\_SWITCH\_SET 22=DODOSIM206FSX.IDLE\_RELEASE\_SET 23=DODOSIM206FSX.TAXI\_LIGHT\_SWITCH\_SET 24=DODOSIM206FSX.TAXI\_LIGHT\_SWITCH\_OFF 25=DODOSIM206FSX.TAXI\_LIGHT\_SWITCH\_ON |

3) Open the FSUIPC.ini configuration file and look for a section called [Events].  If it isn't there, create one and enter the following data:

|  |
| --- |
| **Code Sample** |
| [EventFiles] 0=DodoSim206FSX |

4) Restart FSX and open the FSUIPC dialog. Press a button to set an event assignment as normal, and look at the event selection list.  You'll notice a new set of bindings at the top, preceded by a : character, i.e.:

|  |
| --- |
| **Code Sample** |
| :DODOSIM206FSX.ELECT\_FUEL\_PUMP1\_SET :DODOSIM206FSX.ELECT\_FUEL\_PUMP2\_SET :DODOSIM206FSX.WARNING\_HORN\_CB\_SET :DODOSIM206FSX.NAV\_LIGHTS\_SET ..etc.. |

5) When mapping a button to these functions, they require a parameter value.  As a rule, to set a switch off, use parameter 0, to set it on use 1.  Multi-position switches, such as the five position difficulty switch use parameters in the range 0-4, etc.

## P3D Installation

The up-shot is that if you copy the DodoSim 206 FSX over from an FSX installation to P3Dv2, it will work, but expect occasional crashes when re-loading aircraft or flights.  There is nothing that can be done about this.  
  
To get the 206 FSX working to this state in P3D, you have to copy over from an FSX installation the following files:  
  
1) The four DodoSim directories from within the /SimObjects/Rotorcraft directory.  
2) The DodoSimFSX206.gau from the /gauges directory.  
3) The fx\_DodoSimEngineFire.fx file in the /effects directory.  
  
Additionally, you will need some resources from the default FSX Bell 206 that the DodoSim 206 FSX uses:  
  
1) Copy the whole /SimObjects/Rotorcraft/Bell206B directory over (to get the full set of external sounds)  
2) Copy the /gauges/Bell\_206B.dll file over (to get the default instruments used in places

POST REASSEMBLY CHECK ESTOP AND DOOR SENSOR  
  
Overall removal order  
Rudders  
Seats  
LH collective  
Stop panel and removed door  
Centre collective  
Remove cyclic weight  
Remove cyclic - clearance tight for console  
  
  
  
Removing Seat Base  
  
Remove Seats  
Remove collective assembly as a whole  
Cut cable ties holding cable assembly underneath   
Remove SB external  
Unplug SB rear speakers  
unscrew SB volume control  
Remove SB external USB cable from cable clamp  
Place all cables in centre bay  
Clear rear Cambridge speakers  
Remove 4 coach screws holding frame  
  
  
Centre pedestal installation  
  
Install pedals first as they provide cantering info  
  
Centre removal  
GPS support  
Radio Panels  
Cover on front panels  
  Screws base  
  Bolts top  
  
If removing front facia  
  Bolt at bottom right  
  Six bolts at top  
  
Top maintenance  
Slid left and right panels straight back and up to extract.  They catch on bolt at the front otherwise  
  
Roof   
  Install switch panel  
  Starboard side- access with Port   
  Slide port in from inside up back forward  
  Position port with bolt  
  Use mirror to locate screw hole  
  
Side ceilings can be removed independently  
  
GPS  
  
Need to install USB to serial driver  
4800 bps 500mS interval  
FSUIPC  
GGA, GSV, RMC, GLL, GSA.  
Note GPS data is not sent if in FSUIPC menu.  
  
Lowrance GLL, GGA, 4800 bps. NMEA Input, NMEA GPS INPUT.  
  
Caused blue screen whel plugging in new GPS lead  
On first boot with new USB hung at post.  Install driver for USB to serial PL-2303  
Need to unplug and plug to recognise.  
  
Intercom  
Cigarette lighter has centre positive at 12 volts  
  
If voltage presented is greater than 9 volts on passenger 3 and 4 output runs nicely down to 5 volts, so using 7805.  
  
Main input not squelched out, recorder in is squelched.  
  
DO NOT USE SOCKETS ON OVERHEAD PANEL - Headset cables will block video.  
  
Drill out heads for attaching centre console to floor, needs to be below top face so electronics will fit.  
  
LED backlighting  
  
White  
-|>-                  500 ohm 15.2mA  
-|>- -|>-           330 ohm (330 a10) 15.5 mA     
-|>- -|>- -|>-    150 ohm (220 a10)  13.2 mA  
  
  
Led down lights  
  
Removed all electronics  
@17.2V 203.7mA  
Cold 193-201mA as they warm up  
So for 19v drop 2v @ 200mA is 10 ohms  
W=V\*A  
=2 \* 0.2  
=0.4w around 0.5watts  
  
@18v 1V @ 200mA = 5 ohms  
W = 1 \* 0.2  
= 0.25 watts  
  
  
  
Code based off  
Bne-DCS-SOIC-cmd-line converter  
Bne-arduino-multiled-receiver  
  
For dodsim vars - run dodosim206fsxexportipc  
Updates 0x2f28 - 8 byte concord nose visor  
0  Engine out  
1  Rotor Low  
2  Fuel Low  
3  Generator Fail  
4  Battery Warm  
5  Battery Hot  
..  
20. TOT  
  
FSUIPC offsets

|  |  |  |  |
| --- | --- | --- | --- |
| COM1 | 034E | 2 | BCD |
| COM1 Standby | 311A | 2 | BCD |
| COM2 | 3118 | 2 | BCD |
| COM2 Standby | 311C | 2 | BCD |
| NAV1 | 0350 | 2 | BCD |
| NAV1 Standby | 311E | 2 | BCD |
| NAV2 | 0352 | 2 | BCD |
| NAV2 Standby | 3120 | 2 | BCD |
|  |  |  |  |

3122 Radio Audio 1 switches

|  |  |
| --- | --- |
| 2-7 | Com1Tx |
| 2-6 | Com2Tx |
| 2-5 | Com RX Both |
| 2-4 | Nav 1 Sound |
| 2-3 | Nav 2 Sound |
| 2-2 | Marker |
| 2-1 | DMF |
| 2-0 | ADF |

A leading '1' is assumed for all frequencies  
SOIC  
":" I between variables in UDP  
  
Need to tie IO Card to script index  
  
SOIC searches for script files from its directory, so if getting blank when editing or unable to find file while debugging, probably in wrong directory.  
  
Debug can be used to set variables to light LEDs.  
  
DO NOT FORGET TO RUN HARDWARE UTILITY TO GETTING WARNING PANEL STATUS FROM DODO.  
  
----\_nm ---\_kt  
  
DME     xxxx\_nm  xxx\_KT  
XPonder xxxx  
  
ADF xxx.x  
  
Markers  
O    Blue  
M    Red  
I      White  
  
Radio  
COM1  COM2  BOTH NAV1 NAV2 MKR DME ADF  
  
  
  
Coding  
Handle TCP disconnect - attempt to reconnect on next UDP packet.  
Monitor avionics status on radio lights  
Cleaned up Arduino code, removed special handling for LCD/OLED 2&7  
  
Arduino (com4) -> arduino cli sender (UDP) -> fsx extractor (UDP) -> SOIC-conv (tcp) -> SOIC SERVER  
  
Electronics  
  
Found OverPros encoders not sending long enough pulses, so using Leo Bodnar encoders that supports encoders in a matrix.  
  
Encoder is a ELMA E37 CT6330  
  
Computer Build - SSD upgrade  
Unplug unneeded USB devices, prevents hangs at reboot  
Copied from 224G to 500G SSD took around 15min  
Still need to increase partition size  
  
  
Rebuild FSX installed, Dodo  
Ran up, no gauges displayed  
Installed FSX acceleration  
Gauges displaying correctly  
Installed Dodo SP2  
Reasked about trusting  
Installed FSUIPC  
Pasted panel.cfg into low skid  
Panel undocked  
Difficulty setting knob underneath red text  
On platform   
Install FSUIPC in P3D 2.2  
SimCor still crashed  
Copied Dodo files as per Hover Controls instructions  
Saved layout - 2nd Param in XML Is screen size  
  
Deleted assignments for CHRC and BU0836X  
  
Used new way of assigning toggles which works well.  
  
Needed to use Toggle Tailwheel in FSX panel to get mute to work  
  
Needed to set stop position in throttle within Dodo control panel  
  
Also needed to map throttle to prop axis and tune in FSUIPC   
  
On test computer downloaded build 3 of GoFlight SW  
  
GF LEDs only on when avionics powered  
Key send to other apps  
LUA  
  
FSx Linda enables the use of additional high count joystick controllers. FSX by default only recognises the first 32 inputs  
  
To install copy to FSX modules folder  
You are able to name stick  
need to validate P3D compatibility   
Config to start minimised when launched  
Radio Audio Input Selector

|  |  |
| --- | --- |
| COM1 | 66463 |
| COM2 | 66464 |
| Both | 66465 |
| Nav1 | 65842 |
| Nav2 | 65843 |
| Marker | 66477 |
| DME1 | 65844 |
| ADF | 65846 |

65637  
65636  
65644  
65645  
  
65638  
65639  
65646  
65647  
  
65642  
65643  
65640  
65641  
  
66437  
66436  
66438  
66440

Switch/Rotary Mappings

|  |  |  |  |
| --- | --- | --- | --- |
| Front Panel | Joystick |  |  |
|  |  |  | F = FSUIPC starts 0  L = Linda starts 1  First Param Press, Second Param Release |
| Caution Light Test | None |  |  |
| Warning Horn Mute | 1/23 | F | Toggle Tailwheel Lock /0 |
| TOT LT Test | 1/22? | F | DODOSIM – Tot Test /1 /0 |
| Altimeter Inc | 1/7 | F | Kohlsman Inc |
| Altimeter Dec | 1/6 | F | Kohlsman Dec |
| VOR1 Left Inc | 1/0 | F | VOR 1 Obi Inc Fast |
| VOR1 Left DEc | 1/1 | F | VOR 1 Obi Dec Fast |
| VOR1 Right Inc | 1/8 | F | Heading Bug Inc Fast |
| VOR1 Right Dec | 1/9 | F | Heading Bug Dec Fast |
| Nav/GPS | 1/11 | F | DODOSIM – GPS Drives NAV 1 /0 /1 |
| Gyro Inc | 1/4 | F | Gyro Drift Inc |
| Gyro Dec | 1/5 | F | Gyro Drift Dec |
| VOR2 OBS Inc | 1/2 | F | VOR 2 Obi Fast Inc |
| VOR2 OBS Dec | 1/3 | F | VOR 2 Obi Fast Dec |
| Fuel Valve Open | 1/10 | F | DODOSIM – Fuel Valve Set /1 |
| Fuel Valve Close | 1/10 | F | DODOSIM – Fuel Valve Set /1 |
|  |  |  |  |
| Audio |  |  |  |
| COM 1 | 3/22 | L | 66463 |
| COM 2 | 3/21 | L | 66464 |
| BOTH | 3/4 | L | 66465 |
| Nav 1 | 3/38 | L | 65842 |
| Nav 2 | 3/5 | L | 65843 |
| MKR | 3/7 | L | 66477 |
| DME | 3/20 | L | 65844 |
| ADF | 3/36 | L | 65846 |
| COM 1 Swap | 3/66 | L | 66372 – errs 3/97 |
| COM 2 Swap | 3/65 | L | FSX: COM2 Radio Swap 0 errs 3/37 |
| COM 1 Major Inc | 4/1 | L | 65637 |
| COM 1 Major Dec | 4/2 | L | 65636 |
| COM 1 Minor Inc | 4/8 | L | 65639 |
| COM 1 Minor Dec | 4/7 | L | 65638 |
| COM 2 Major Inc | 4/19 | L | 66437 |
| COM 2 Major Dec | 4/20 | L | 66436 |
| COM 2 Minor Inc | 4/22 | L | 66440 |
| COM 2 Minor Dec | 4/21 | L | 66438 |
| Nav 1 Swap | 4/67 | L | FSX NAV 1 Radio Swap err 4/51 |
| Nav 2 Swap | 4/52 | L | FSX NAV 2 Radio Swap |
| Nav 1 Major Inc | 4/16 | L | 65641 |
| Nav 1 Major Dec | 4/15 | L | 65640 |
| Nav 1 Minor Inc | 4/14 | L | 65643 |
| Nav 1 Minor Dec | 4/13 | L | 65642 |
| Nav 2 Major Inc | 4/4 | L | 65645 |
| Nav 2 Major Dec | 4/3 | L | 65644 |
| Nav 2 Minor Inc | 4/10 | L | 65647 |
| Nav 2 Minor Dec | 4/9 | L | 65646 |
| AP HDG Inc | 4/6 | L | Unassigned |
| AP HDG Dec | 4/5 | L | Unassigned |
| AP V/S Inc | 4/29 | L | Unassigned |
| AP V/S Dec | 4/30 | L | Unassigned |
| AP CRS Inc | 4/12 | L | Unassigned |
| AP CRS Dec | 4/11 | L | Unassigned |
| AP Alt Inc | 4/23 | L | Unassigned |
| AP Alt Dec | 4/24 | L | Unassigned |
| AP IAS Inc | 4/18 | L | Unassigned |
| AP IAS Dec | 4/17 | L | Unassigned |
| AP CRS | 3/3 | L | Unassigned errs 3/16 |
| AP HDG | 3/17 | L | Unassigned errs 3/33 |
| AP IAS | 3/18 | L | Unassigned errs 3/34 |
| AP ALT | 3/19 | L | Unassigned errs 3/35 |
| AP APPR | 3/49 | L | Unassigned |
| AP Engage | 3/18 | L | Unassigned |
| AP FD | 3/35 | L | Unassigned errs 3/51 |
| AP BC | 3/50 | L | Unassigned |
| Ident 1 | 2/79 | L | FSX ATC Menu 1 |
| Ident 2 | 2/78 | L | FSX ATC Menu 2 |
| Ident 3 | 2/80 | L | FSX ATC Menu 3 |
| Ident 4 | 2/127 | L | FSX ATC Menu 4 |
| Ident 5 | 2/126 | L | FSX ATC Menu 5 |
| Ident 6 | 2/128 | L | FSX ATC Menu 6 |
| Ident 7 | 2/111 | L | FSX ATC Menu 7 |
| Ident 8 | 2/110 | L | FSX ATC Menu 8 |
| Ident 9 | 2/112 | L | FSX ATC Menu 9 |
| Ident \* | 2/95 | L | FSX Pause Toggle |
| Ident 0 | 2/94 | L | FSX ATC Menu 0 |
| Ident # | 2/96 | L | FSX ATC –Brings up ATC Window |
| ADF Major Inc | 4/32 | L | FSX ADF 100 Inc |
| ADF Major Dec | 4/31 | L | FSX ADF 100 Dec |
| ADF Minor Inc | 4/26 | L | FSX ADF 1 Inc (this is due to a lack of ADF rotaries) |
| ADF Minor Dec | 4/25 | L | FSX ADF 10 Dec |
| Gear Up | 3/113 | L |  |
| Gear Down |  | L |  |
| Flaps Up | 3/81 | L | Errs 3/98 3/113 |
| Flaps Down |  |  |  |
| Speed Brake Extend | 3/98 | L | Errs 3/114 |
| Speed Brake Retract |  |  |  |
| Brake | 3/83 |  |  |
| Caution Lights Dim |  |  |  |
| Caution Lights Bright |  |  |  |
| Eng Deice Off | 3/66 |  | Errs 3/83 3/98 |
| Eng Deice On |  |  |  |
| Hydral Off | 3/81 |  | Errs 3/98 3/83 |
| Hydra On |  |  |  |
| Stick Top Button | 0/7 |  | (need to clear cycle view) |
| Stick Front Button | 0/6 |  | Perhaps ATC Menu 1 |
| Collective Lights Fwd | 0/4 | F | Landing Lights Set /1 /0 |
| Collective Lights Both | 0/5 | F | Landing Lights Set /2 /1 |
| Collect Lights Off |  |  |  |
| Coll Gov Inc | 0/2 | F | Increase Helo Beep /0 /0 Errs 0/32 |
| Coll Gov Dec | 0/3 | F | Decrease Helo Beep /0 /0 |
| Collective Starter | 0/0 |  |  |
| OH Start | 3/28 | F | DODOSIM Generator Switch /1 /0 |
| OH Gen |  |  |  |
| OH Batt Off |  |  |  |
| OH Bat On | 3/27 | F | DODOSIM Master Battery Set /1 /0 |
| OH Avionics Off |  |  |  |
| OH Avionics On |  |  |  |
| OH Pitot Heat Off |  |  |  |
| OH Pitot Heat On |  |  |  |
| Dir Gyro Off |  |  |  |
| Dir Gyro On | 3/26 | F | Dir Gyro Set /1 /0 |
| Diff 1 | INOP |  |  |
| Diff 2 | 3/7 | F | DODOSIM Difficulty /2 |
| Diff 3 | 3/8 | F | DODOSIM Difficulty /3 |
| Diff 4 | 3/9 | F | DODOSIM Difficulty /4 |
| Diff 5 | 3/10 | F | DODOSIM Difficulty /5 |
| Heat |  |  |  |
| Heat – off |  |  |  |
| Heat – Vent on | 3/11 | F | DODOSIM Hydraulic Set /1/0 |
| Heat - Heat | 3/12 |  |  |
| GEN Field Close | 3/24 | F | DODOSIM Gen Field Set /1 /0 |
| GEN Field Open |  |  |  |
| GEN Reset Close | 3/25 | F | DODOSIM Gen Reset Set /1 /0 |
| GEN Reset Open |  |  |  |
| Pos Light Off |  |  |  |
| Pos Light On |  |  |  |
| Anti-Coll Light off |  |  |  |
| Anti-Coll Light On |  |  |  |
| Fuel Boost Aft Close | 3/23 | F | DODOSIM Fuel Pump 2 /1 /0 |
| Fuel Boost Aft Open |  |  |  |
| Fuel Boost Fwd Close | 3/22 | F | DODOSIM Fuel Pump 1 /1 /0 |
| Fuel Boost Fwd Open |  |  |  |
|  |  |  |  |

Todos  
  
Separate collective from electronics - look at db15s for all connectors then can swap  
Cable pedals from under floor  
Explore mat options  
Update document  
Paint black bolt heads  
  
Approach  
Ceiling down  
Paint outer back first  
Sanding with vacuum  
Roof consider different color from outer  
  
  
Determine how to power down lights consider replacing PCB WITH RED/white array running @ 12v with internal resistors  
  
Post completion look at 3d printing coupler for flight link control heads  
Cut circles out to plug overhead headphone socket locations laser out centres to run bolt through  
  
Test throttle assembly with KingAir  
  
Find switch with screw able top for landing light  
Mouser 7105TZQE  
Mom-on-mon flaps and speed brake  
Consider how to mount cougar throttle and stick  
  
Longer term todos  
  
Radios using Pete Bodnar encoders  
Removal of JetRanger physical panel overlay for other aircraft

# Sound System

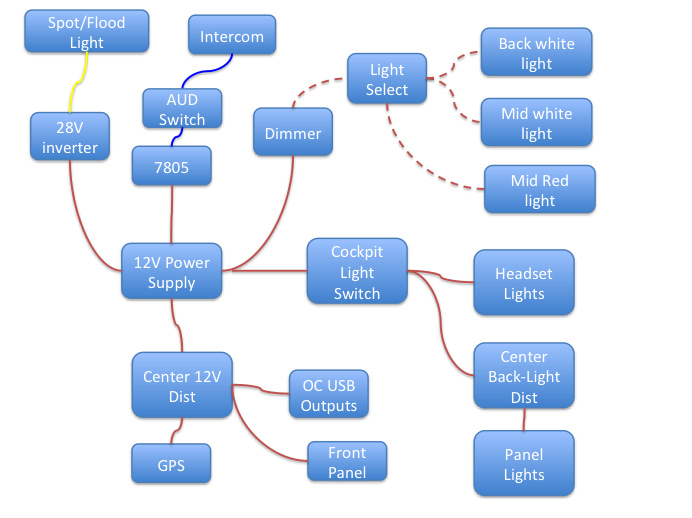
PILOT PA400S Stereo 4 Place Portable Intercom

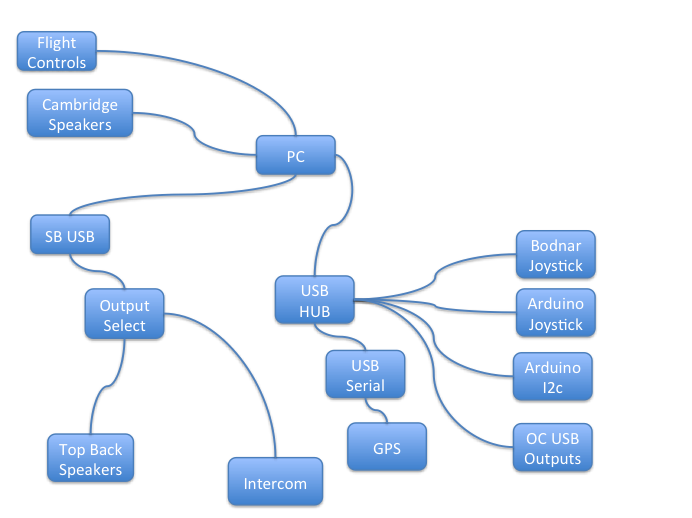
The audio in/out cable, when plugged into the "recorder out" socket allows the intercom audio, including any ATC transmissions, to be recorder on a suitable recording device. When plugged into the "recorder in" socket, the audio in/out cable allows the playing of stereo music or other audio entertainment over the intercom audio.

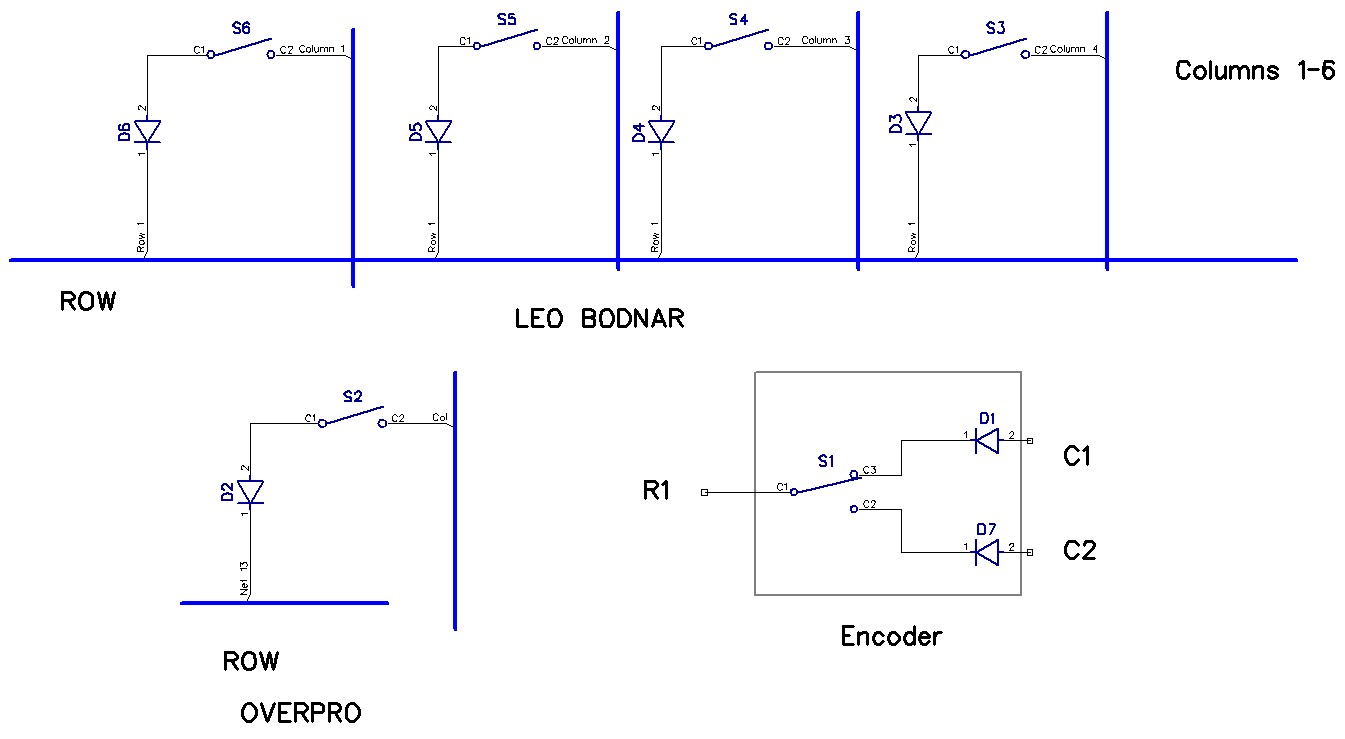
The on/off press button to the right of the squelch knob is used to turn on and off the red back lighting for use at night. The back lighting should be left off whenever practical as it consumes battery power.

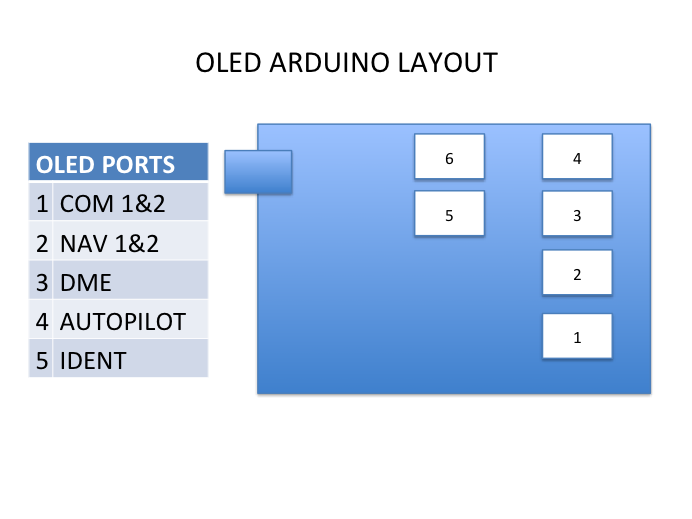
Please note, that as an added battery saving feature, the PA400S is automatically switched off when the pilot’s phone plug is removed.

Distributed by www.pilotcommunications.com.au, Ozcomm Australia









# FSX Data flows for radios and lights

# 

# Data Flows

USB I/O 1

USB I/O 2

USB I/O 3

CDU

Brydling

Leo Bodnar

Arduino

I2C Splitter

GPS

Oled 1

Oled 2..

Oled ..6

Oled 7

Dimmable

Simmeters

Gauges

# Data Flows with Protocol and Ports

USB I/O 1

USB I/O 2

USB I/O 3

CDU

Brydling

Leo Bodnar

Arduino

I2C Splitter

GPS

Oled 1

Oled 2..

Oled ..6

Oled 7

Dimmable

Simmeters

Gauges

TCP-8092

UDP -7777

UDP -7780

UDP -7781

UDP -7782

COM4

Memory Scan

UDP 6061

UDP -7780

SlimDX

SlimDX

COM?

DX

DX

UDP -7781

# Devices

12V-24V

Inverter

12V LED - Relay

Fire Handle - 1

Magnetic

Switch – 1

L - Pitch

Magnetic

Switch – 2

R - Pitch

Magnetic

Switch – 3

L - Yaw

Magnetic

Switch – 6

EAC

Magnetic

Switch – 5

Anti-Lock

Magnetic

Switch – 4

R - Yaw

Fire Handle - 2

Fire Handle - 3

Landing Gear Light - 1

Engine Cluster

Altimeter

VSI

ADI

HSI

Flaps

Flood-lighting

Control - A

Back-lighting

Control - A

Rudder Pedals

I2C Demux

2 \* 16 OLED

VHF-AM

2 \* 16 OLED

VHF FM

2 \* 16 OLED

UHF

2 \* 16 OLED

ILS

2 \* 16 OLED

CSMC

2 \* 20 OLED

CSMP

2 \* 16 OLED

TACAN

Arduino

Thrustmaster

Warthog

OC CDU

Leo Bodnar BBI-32

Brydling

OC USB Output -3

OC USB Output -2

OC USB Output -1

Non-Dimmable

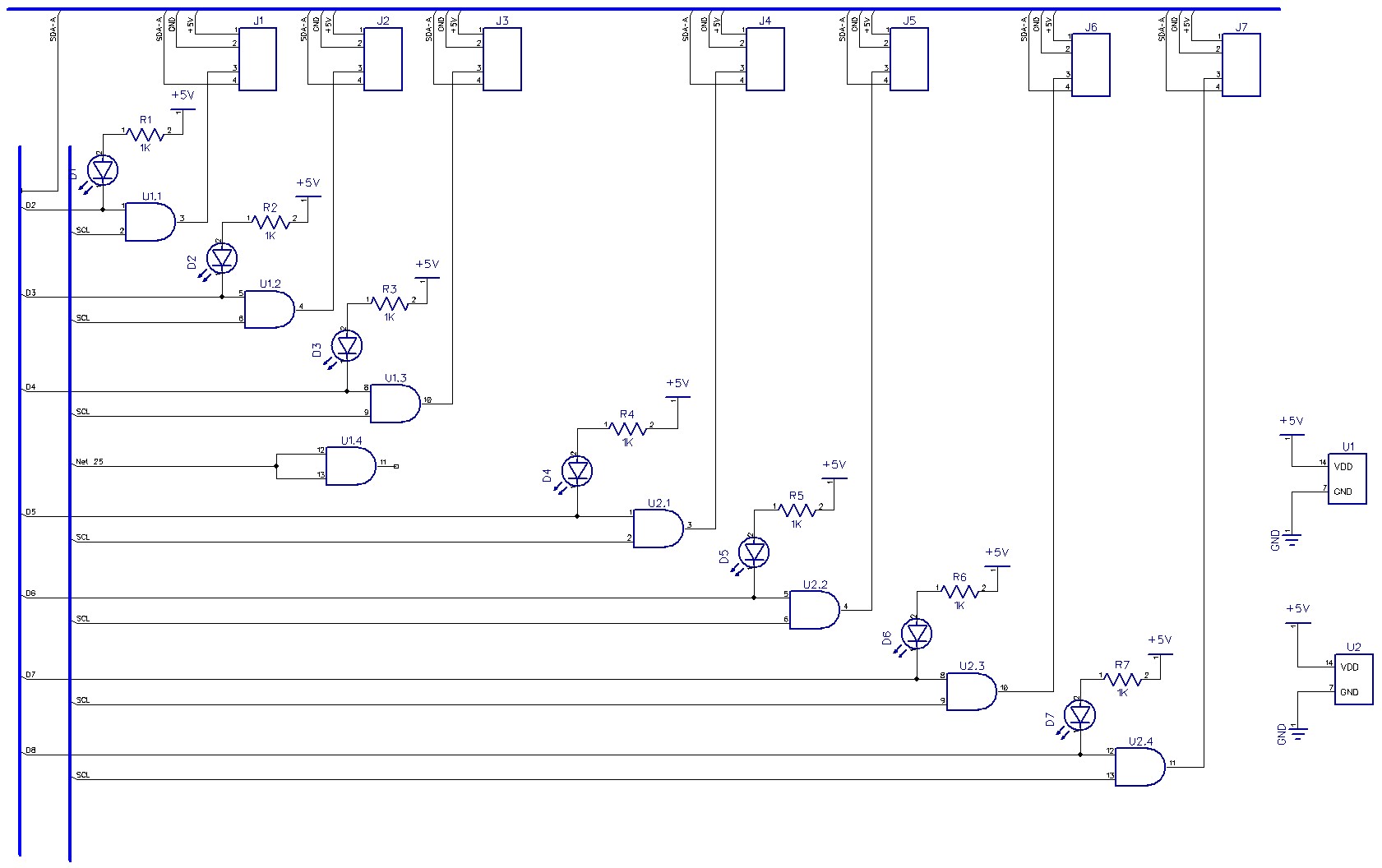
Thrustmaster

R-MFD

Thrustmaster

L-MFD

# I2C Demux Schematic



1. Starts at 0 [↑](#footnote-ref-1)