

Board with 25 LEDs and 8 switches

13 LEDs for annunciator (8 yellow, 5 red)

3 LEDs for OMI (blue, red, white)

7 LEDs for 7 eSwitch 5501 switches (green LEDs)

2 LEDs for 1 eSwitch 5501 switch (RED/GREEN LEDs)

LEDs driven by MAX7219, supported by 1-20K resistor, 1-10uf cap, 1-.01uf cap

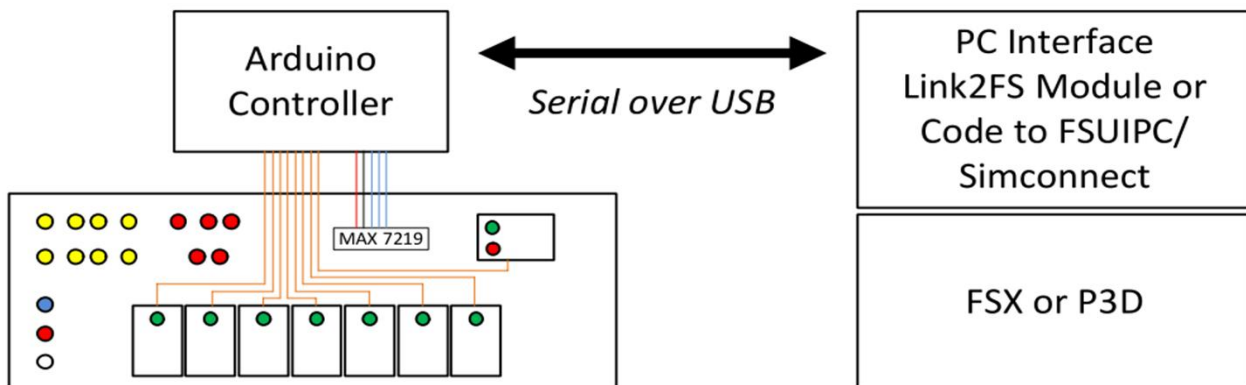
Interface Software on PC (Arduino talks to PC via USB serial commands)

Link2FS  
or FSUIPC

<http://www.jimspage.co.nz/intro.htm>

Interface code in VB.NET 2013 with Paul Henty's FSUIPC client 2.4

General Sketch



This project needed a bunch of LEDs, 25 total in fact: 7 for buttons, 2 for GPS/NAV indicator, 3 for OMI indicators, and 13 for the annunciator.

LED Driving Problem: Any given AVR microcontroller pin is allowed to source up to 40ma, and the total power draw needs to be below 200ma for entire device. 25 LEDs could draw up to  $25 \times 20\text{ma} = 500\text{ma}$ , way more than an Arduino could source. And, I want the whole project to be driven from a single USB connection and no external power supply.

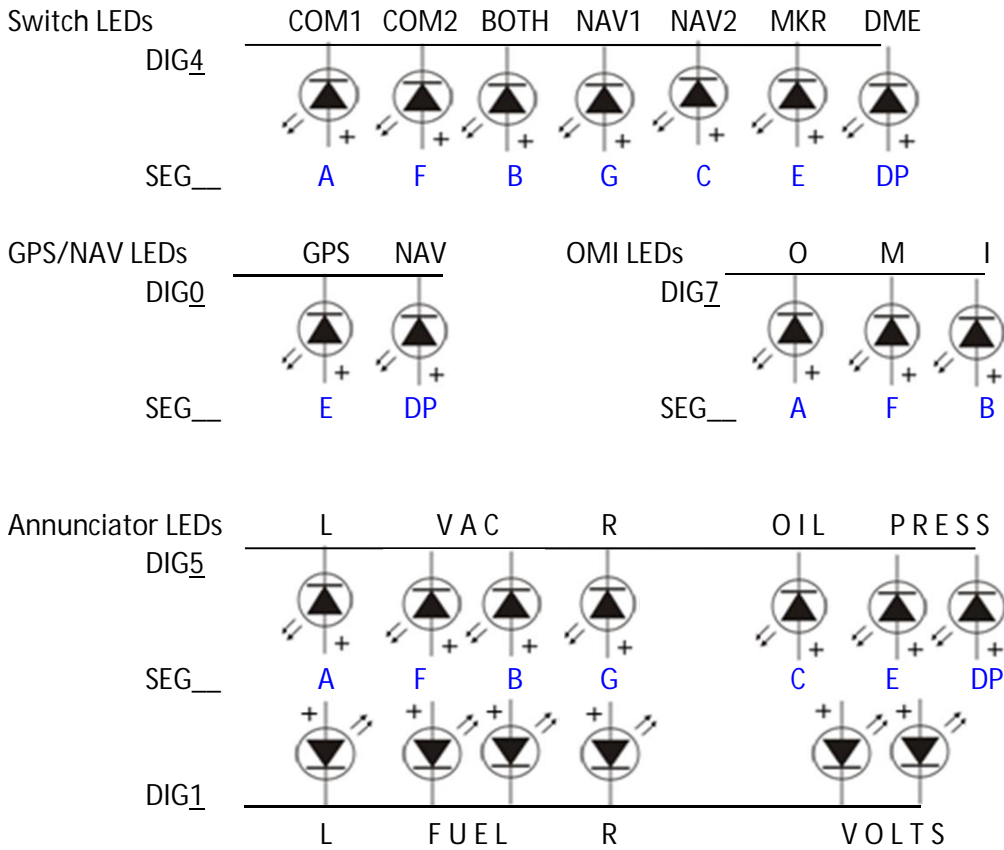
LED Driving Solution: Use the handy [MAX7219 chip](#). It's intended to drive up to 8 digits of seven segment displays, but can just as easily drive up to 72 single LEDs. And, it takes just 3 control pins from an Arduino (plus 5v and GND) and uses only 3 external components (2 capacitors and 1 resistor). The MAX7219 "strokes" the LEDs, flickering them on and off in sequence such that the total power required at any one time can be considered just one LED (20ma). This solution solves the overall power issue and reduces wiring complexity.

Switches: The [eSwitch 5501 series](#) seems like a good fit, it's a switch with built in LED.

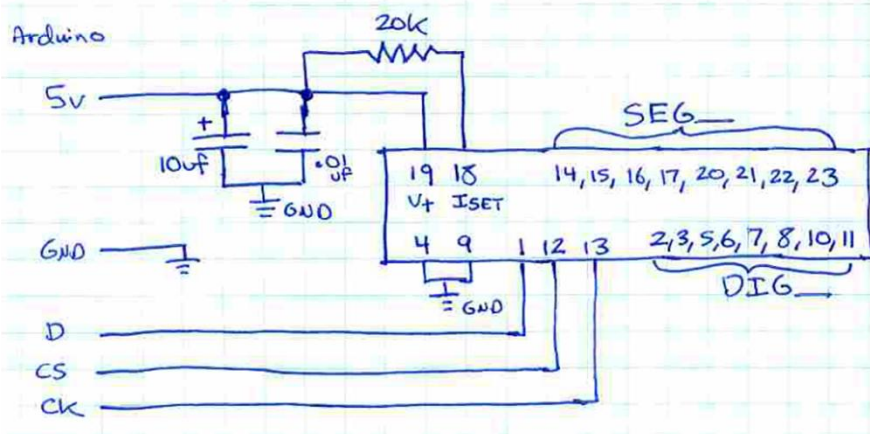
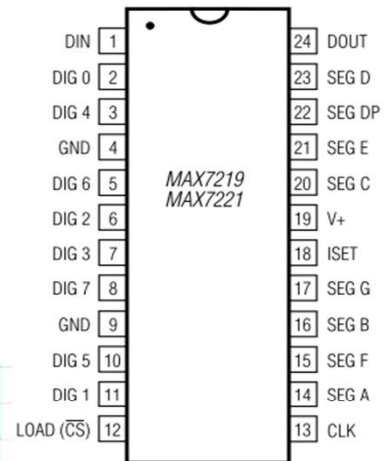
Annunciator: This was a little tricky, I wanted the words like "VOLTS" to light up behind a dark panel. After some trial and error, I found that putting a row of red or yellow LEDs behind a smoke

Project and assembly pictures:

<http://www.simav8.com/>



DIG0	GPS/NAV LEDs
DIG1	6 Annunciator LEDs (bot row)
DIG2	n/a
DIG3	n/a
DIG4	7 switch LEDs
DIG5	7 Annunciator LEDs (top row)
DIG6	n/a
DIG7	3 OMI LEDs



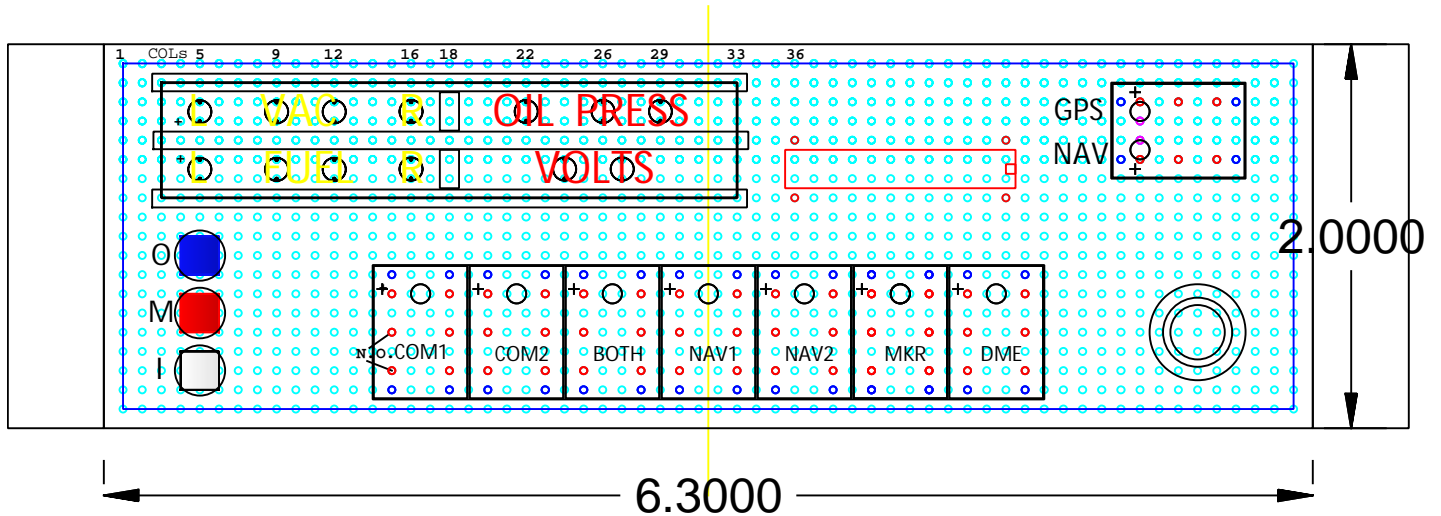
SIMComm Build 20161124.dwg

drawing: Standard.dwt, grid .1, snap .1 or .025

7x eSwitch 5500 series, single GRN  
1x eSwitch 5500 series, dual GRNRED  
No knob used  
OMI lights - ink pen caps with leds behind  
Indicator panel, smoke acrylic with boxes  
behind to make four cavities

25 led total, drive with MAX7219

NOTE: no offset grid! Starts at 0 this project  
because the switches are even 10ths



Print this on inkjet transparency paper  
to make the annunciator overlay

(this page rendered in PDF at 600dpi)

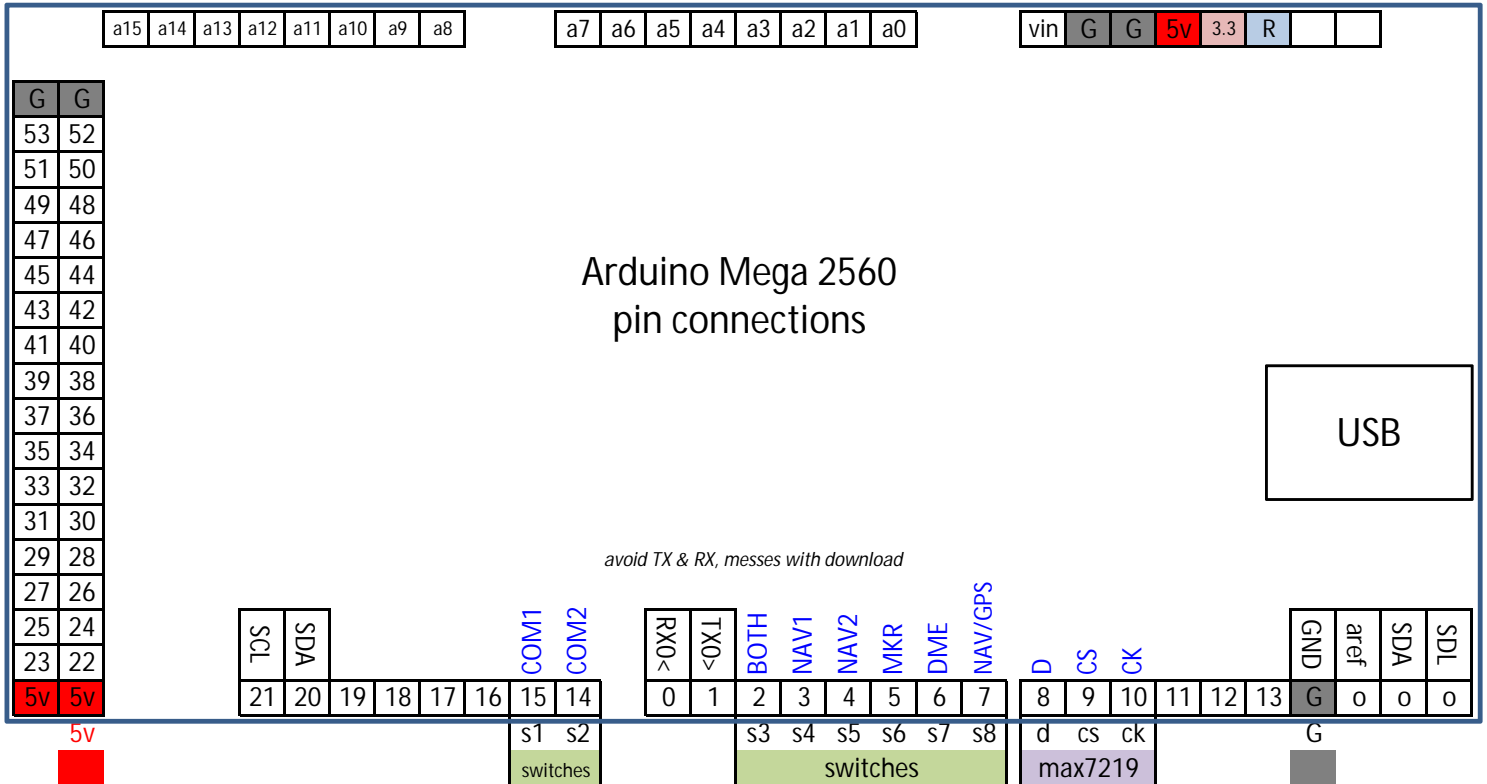
1 COLs 5 9 12 16 18 22 26 29 33 36

L	VAC	R	OIL PRESS
L	FUEL	R	VOLTS

GPS  
NAV

O  
M  
I

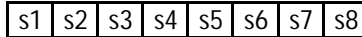
+  
No.COM1 COM2 BOTH NAV1 NAV2 MKR DME



header, Max7219 (5v closest to CAPS)



header, switches (bot of board = s1)



Max7219 SEG bits

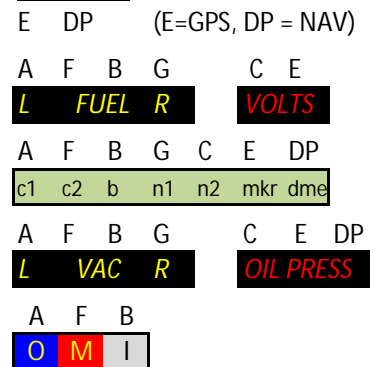
```
lc.setRow(disp,dig,B01000000); // A
lc.setRow(disp,dig,B00000010); // F
lc.setRow(disp,dig,B00100000); // B
lc.setRow(disp,dig,B00000001); // G
lc.setRow(disp,dig,B00010000); // C
lc.setRow(disp,dig,B00000100); // E
lc.setRow(disp,dig,B10000000); // DP
lc.setRow(disp,dig,B00001000); // D
lc.setRow(disp,dig,B00000000); // off
```

DIG

Function

- DIG0 switch GPS/NAV LED (2)
- DIG1 bot row leds (6)
- ~~DIG2 n/a~~
- DIG4 switch LEDs (7, s1-s7)
- ~~DIG3 n/a~~
- DIG5 top row leds (7)
- ~~DIG6 n/a~~
- DIG7 OMI leds

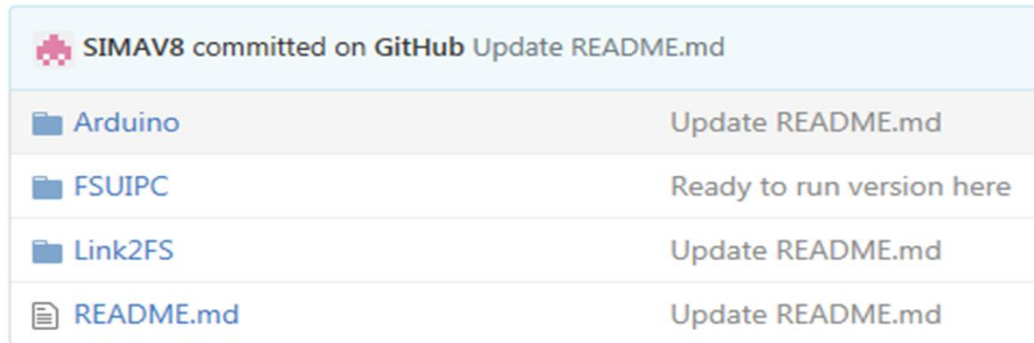
SEGments



## Arduino firmware and PC interface software

It's all available, for free! Full project source available on

<https://github.com/SIMAV8/SIMCMP1>



SIMAV8 committed on GitHub Update README.md	
Arduino	Update README.md
FSUIPC	Ready to run version here
Link2FS	Update README.md
README.md	Update README.md

### Arduino Firmware

Available, nothing hidden, it's all right there for you to use!

### PC interface Software (2 methods)

Link2FS: The Arduino can be driven by easily configurable tool Link2FS available from Jim <http://www.jimspage.co.nz/intro.htm>

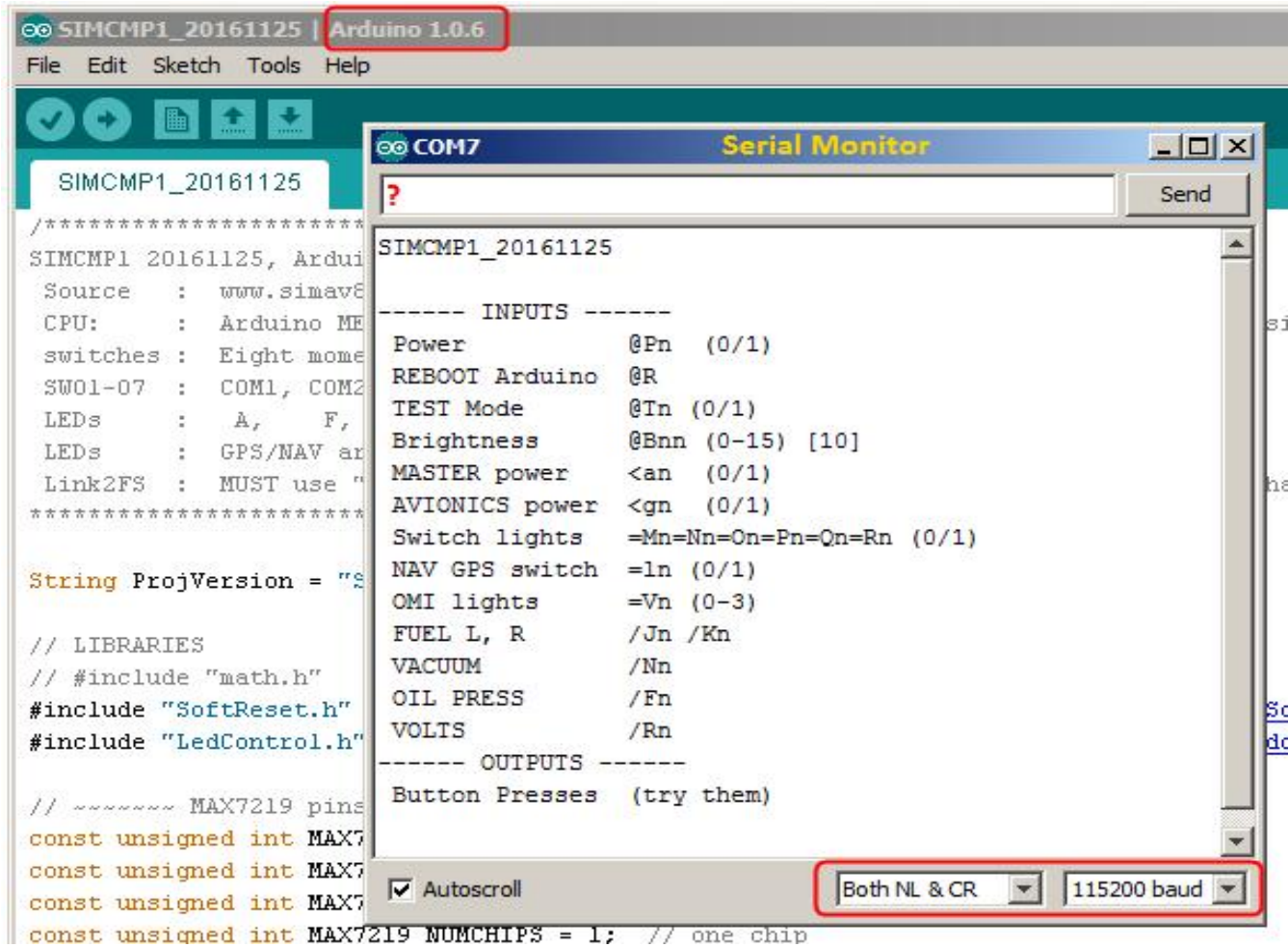
Config for SIMCMP1, see the Link2FS folder at github

FSUIPC: you can use (or customize) a direct interface to FSUIPC written in VB.net 2013  
Source code and information in the FSUIPC folder of github

Windows EXE: A standalone (compiled) version of the VB code is available under [FSUIPC / SIMcmp1\\_fsuipc / bin / Release /](#) at github

SIMAV8 SIMCMP1 SFT Cmds, 20161113			Link2FS	Link2fs Multi FSX v6f.exe	<a href="http://www.jimspage.co.nz/intro.htm">http://www.jimspage.co.nz/intro.htm</a>
<a href="http://www.SIMAV8.com">www.SIMAV8.com</a>			FSUIPC	VB.NET 2013 via Paul Henty's interface v2.4	<a href="#">FSUIPC Client DLL for .NET v2.4</a>
OUTPUTS (from Arduino)					
	TYPE	ARDUINO	Link2FS	FSUIPC	<i>future</i>
COM1	button	A45	A45	&H3122, b7	SIMCONNECT
COM2	button	A46	A46	&H3122, b6	KEY_COM1_TRANSMIT_SELECT
BOTH	button	A47	A47	&H3122, b5	KEY_COM2_TRANSMIT_SELECT
NAV1	button	A48	A48	&H3122, b4	KEY_COM_RECEIVE_ALL_TOGGLE (or KEY_COM_RECEIVE_ALL_SET)
NAV2	button	A49	A49	&H3122, b3	KEY_RADIO_VOR1_IDENT_SET (on / off)
MKR	button	A53	A53	&H3122, b2	KEY_RADIO_VOR2_IDENT_SET (on / off)
DME	button	A50	A50	&H3122, b1	KEY_MARKER_SOUND_SET (may not work)
GPSNAV	button	A54	A54	&H132C	KEY_RADIO_DME1_IDENT_SET
					KEY_TOGGLE_GPS_DRIVES_NAV1
INPUTS (to Arduino)					
TestAllLEDs		@Tx			<u>Comments</u>
s1 = COM1	LED	=Mx	=Mx	&H3122, b7	<u>Max7219 DigX SegY</u>
s2 = COM2	LED	=Nx	=Nx	&H3122, b6	@T1 = test all, @T0 = normal
s3=BOTH	LED	=Ox	=Ox	&H3122, b5	LED Off or 1n
s4=NAV1	LED	=Px	=Px	&H3122, b4	LED Off or 1n
s5=NAV2	LED	=Qx	=Qx	&H3122, b3	LED Off or 1n
s6=MKR	LED	=Ux	=Ux	&H3122, b2	LED Off or 1n
s7=DME	LED	=Rx	=Rx	&H3122, b1	LED Off or 1n
s8=GPSNAV	LED	=lx (lima)	=lx	&H132C	LED Off or 1n
					LED GPS or NAV
FUEL_L	LED	/J1	/J1	&HB7C	Left Fuel level (low < 030)
FUEL_R	LED	/K1	/K1	&HB94	Right Fuel level (low < 030)
VAC_L	LED	/N1	/N1	&HB18	Left VAC low
VAC_R	LED	/N1	/N1		Right VAC low
OIL_PRESS	LED	/F1	/F1	&H3B60	Oil Pressure level (low < 21)
O_M_I	LED	=Vn	=Vn	&HBAC, AE, B0	MKR (0=None, 1=Out, 2=Mid, 3=In)
VOLTS	LED	/R1	/R1	&H2840	VOLTS low (< 0)
		<an	<an	&H2840	MasterBusVolts > 12 = <a1 else <a0
volts (avionics)	LED	<gn	<gn	&H2850	AvionicsBusVolts > 12 = <g1 else <g0
					Note: <a1 powers annunciators
					Note: <g1 powers COM panel/OMI

At the Arduino level, you can launch the Serial Monitor and send commands to the hardware



In the monitor, you should immediately get the header string of "SIMCMP1\_{date}"  
Send it a question mark and you'll get a display of possible actions.

Put it in test mode (light up all the LEDs), issue command @T1 (turn back off with @T0)  
Or, press and hold COM1 and DME buttons for 4 seconds

Turn on COM1 and DME switch LEDs, issue commands:

- <a1 simulate MASTER being on (other commands won't work if not!)
- <g1 simulate AVIONICS being on (same as above)
- =M1 turn on COM1 light
- =R1 turn on DME light
- /R1 light up VOLTS annunciator
- =M0 turn off COM1 light
- <g0 simulate AVIONICS power turn off (DME light goes out, VOLTS stays on)
- <a0 simulate MASTER power off (VOLTS goes out too)